Optimal Control of Hybrid Systems with Regional Dynamics

A Thesis
Presented to
The Academic Faculty

by

Angela Schöllig

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Engineering Science and Mechanics

School of Civil and Environmental Engineering
Georgia Institute of Technology
December 2007

Optimal Control of Hybrid Systems with Regional Dynamics

Approved by:

Dr. Magnus Egerstedt, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Anthony Joseph Yezzi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Laurence J. Jacobs
School of Civil and Environmental Engineering
Georgia Institute of Technology

Date Approved: August 24, 2007

To my family.

ACKNOWLEDGEMENTS

Let us be grateful to people who make us happy;
they are the charming gardeners who make our souls blossom.
Thank you.

-Marcel Proust

First of all, I would like to express my sincere gratitude to Prof. Laurence J. Jacobs. His great dedication for the exchange program between the University of Stuttgart, Germany, and the Georgia Institute of Technology gave me the unique opportunity to experience studying, research, and life abroad and to broaden my horizon in many different ways. I cannot thank him enough for his generosity, understanding, support, and advice.

I greatly appreciate and value the guidance and support of my advisor Prof. Magnus Egerstedt. His enthusiasm, ingenuity, and optimism were very encouraging and motivating during my research in his lab. I admire his creative and straightforward work. Moreover, he gave me a better understanding of the "academic world" and the great opportunity to work together with Prof. Peter E. Caines from the McGill University in Montréal, Canada. I thank him for interesting and fruitful discussions and his kind and patient cooperation, especially a few hours before the submission deadline. Furthermore, I would like to thank Prof. Anthony Yezzi for serving on my reading committee.

I am pleased to thank my fellow students in the GRITS Lab for their friendship, many enjoyable conversations, and their moral support in the last days of my work on this thesis. I will miss you! Of course, I will also never forget all the friends I met Larry's group in the civil engineering department.

In Germany, I want to thank Prof. Dr.-Ing. Gaul as the head of the ISAP

Exchange Program. The generous fundings of the German Academic Exchange Service (DAAD) and the National Academic Foundation Cusanuswerk are gratefully acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BVP	Boundary Value Problem
DP	Dynamic Programming
DSC	Discrete System Control
HBE	Hybrid Bellman Equation
HOCP	Hybrid Optimal Control Problem
MPTPP	Multimodal Point-To-Point Problem
PMP	Pontryagin's Minimum Principle
TPBVP	Two-Point Boundary Value Problem

LIST OF SYMBOLS

$\mathbb{N}$	Set of all positive integers $\mathbb{N} = \{1, 2, 3, \dots\}$, *page 7*
$\mathbb{R}$	Set of all real numbers, *page 5*
n, m	Positive integers $n, m \in \mathbb{N}$, *page 5*
k, k_0, k_f	Non-negative integers $k, k_0, k_f \in \mathbb{N} \cup \{0\}$, where $k_0 \leq k_f$, *page 7*
A'	Transpose of a vector or a matrix $A \in \mathbb{R}^{n \times m}$, *page 16*
A^{-1}	Inverse of a matrix $A \in \mathbb{R}^{n \times n}$, *page 19*
$A > 0$	Positive definite (symmetric) matrix $A \in \mathbb{R}^{n \times n}$, *page 18*
$A \geq 0$	Positive semi-definite (symmetric) matrix $A \in \mathbb{R}^{n \times n}$, *page 18*
$I_{n \times n}$	Identity matrix $I_{n \times n} \in \mathbb{R}^{n \times n}$, *page 20*
$\overline{\mathcal{S}}$	Closure of $\mathcal{S}$, *page 5*
$\partial \mathcal{S}$	Boundary of an open set $\mathcal{S}$, *page 28*
$\mathcal{U}_{ad}$	Set of admissible input values, *page 7*
$\mathcal{U}$	Set of all bounded measurable functions on the interval $[0, T]$, $T < \infty$ taking values in the set U, $\mathcal{U} = \mathcal{U}(U, L_\infty([0, T]))$, *page 34*
$\mathcal{I}$	Set of positive integers specifying the regions D_i, $i \in \mathcal{I} = \{1, 2, \dots, q\}$, where q denotes the total number of regions, *page 33*
$\mathcal{Q}$	Discrete state space $Q = \{q_i : \ i \in \mathcal{I}\}$ of the regional dynamics system, *page 36*
$\mathcal{E}$	Finite set of transition labels, also referred to as events, *page 58*
$\mathcal{E}^*$	Set of all finite strings of elements in $\mathcal{E}$ including the empty string ϵ, *page 65*

$m_{(i,j)}$	Switching manifold, $m_{(i,j)} = \partial D_i \cap \partial D_j, \quad i,j \in \mathcal{I}$, *page 55*
$(\cdot)^d$	The variable $(\cdot)$ takes values from a finite set of elements, *page 41*
$\frac{\partial F}{\partial x}(x)$	Gradient of the scalar function $F(x) \in \mathbb{R}$, *page 16*
$\frac{\partial f}{\partial x}(x)$	Jacobian of the vector-valued function $f(x) \in \mathbb{R}^m$, *page 16*
J	Performance measure, also called cost functional, *page 5*
J^*	Value function, the cost functional associated with an optimal control input u^*, $J^* = J^*(u^*)$, *page 6*
V	Cost-to-go function, *page 9*
e	Discrete control input, $e \in \mathcal{E}$, *page 58*
$u(\cdot)$	Input signal of a continuous-time dynamical system, $u(\cdot)$ denotes the segment $u(\tau) : \tau \in [t_0, t_f]$, *page 6*
$\{u(k)\}$	Input sequence of a discrete-time dynamical system, $\{u(k)\}$ denotes the sequence $\{u(k_0), \ldots, u(k_f - 1)\}$, *page 7*
u^*	Optimal control input, *page 6*
w	String of discrete control inputs $w = e_1 e_2 \ldots e_M; e_i \in \mathcal{E}, i \in \{1, 2, \ldots, M\}$, $M \in \mathbb{N}$, *page 60*
w^*	Optimal discrete control input sequence $w^* = e_1 e_2 \ldots e_{M^*}$; $e_i \in \mathcal{E}$, $i \in \{1, 2, \ldots, M^*\}$, *page 76*
$S(\overline{\tau}, w)$	Discrete control sequence, *page 60*
$S^*(\overline{\tau}, w)$	Optimal discrete control sequence, *page 60*
x^*	Optimal state trajectory associated with an optimal input u^*, *page 9*
x_k	Continuous state $x_k \in \mathcal{X}$ subjected to the dynamic regime $\dot{x}_k = f_k(x_k, u), \ k \in \mathcal{I}$, *page 35*

$x_{i,k}$	The kth state-component of agent i, $x_{i,k} \in \mathbb{R},\ i \in \{1, 2, \ldots, N_A\},\ k \in \{1, 2, \ldots, n\}$, *page 105*
ξ_0	Given starting point of a trajectory, *page 6*
ξ_f, ξ_T	Given end point of a trajectory, *page 6*
ξ_s^m	State of the mth switch along a hybrid trajectory $x(\cdot)$, *page 24*
$(\xi_s^m)^*$	State of the mth switch along a hybrid *optimal* trajectory $x^*(\cdot)$, *page 24*
t_s^m	Time of the mth switch along a hybrid trajectory $x(\cdot)$, *page 24*
$\overline{\tau}$	Sequence of switching times, where $0 < t_s^1 < t_s^2 < \cdots < t_s^M < T$, *page 60*
$(t_s^m)^*$	Time of the mth switch along a hybrid *optimal* trajectory $x^*(\cdot)$, *page 24*
$(t_s^m, \xi_s^m)^*$	Time-state pair of the mth switch along a hybrid *optimal* trajectory $x^*(\cdot)$, *page 23*
N	Given upper bound $N \in \mathbb{N}$ on the total number of switches along a hybrid trajectory, *page 26*
N_A	Total number of robots in a given multi-agent system, *page 105*
M^*	Optimal number of switches $M^* \in \mathbb{N} \cup \{0\}$ associated with a hybrid optimal trajectory $x^*(\cdot)$ assuming a given upper bound $N \in \mathbb{N}$ on the total number of switches, *page 26*
L_M	Subset of $\mathcal{E}^*$ containing all such strings $e \in \mathcal{E}^*$ which consist of exactly M events, *page 66*
$L(A)$	Language generated by an automaton A, *page 65*
$L_m(A)$	Marked language of an automaton A, *page 66*
$L_m(M, A)$	Subset of $L_m(A)$ containing all such strings $e \in L_m(A)$ which consist of exactly M events, $L_m(M, A) = L_m(A) \cap L_M$, *page 66*

$F_K(M, A)$ Set of all suffices of $L_m(M, A)$ which consist of exactly K events, $F_K(M, A) = L_K \cap \mathit{suff}(L_m(M, A))$, *page 66*

$\mathit{suff}(L)$ Suffix closure of a language L, *page 66*

$\mathcal{G}$ Graph $\mathcal{G} = (V, E)$ corresponding to the set of nodes V and the set of edges E, *page 106*

$A(\mathcal{G})$ Adjacency matrix associated with the graph $\mathcal{G}$, *page 106*

$D(\mathcal{G})$ Degree matrix associated with the graph $\mathcal{G}$, *page 107*

$\mathcal{L}(\mathcal{G})$ Graph Laplacian associated with the graph $\mathcal{G}$, $\mathcal{L}(\mathcal{G}) = D(\mathcal{G}) - A(\mathcal{G})$, *page 107*

κ Leader vector, *page 109*

SUMMARY

In this work, *hybrid systems with regional dynamics* are considered. These are systems where transitions between different dynamical regimes occur as the continuous state of the system reaches given switching surfaces. In particular, the attention is focused on the optimal control problem associated with such systems. More precisely, given a specific cost function, the goal is to determine the optimal path of going from a given starting point to a fixed final state during an *a priori* specified time horizon.

The key characteristic of the approach presented in this thesis is a hierarchical decomposition of the hybrid optimal control problem, yielding to a framework which allows a solution on different levels of control. On the highest level of abstraction, the regional structure of the state space is taken into account and a discrete representation of the connections between the different regions provides global accessibility relations between regions. These are used on a lower level of control to formulate the main theorem of this work, namely, the *Hybrid Bellman Equation* for multimodal systems, which, in fact, provides a characterization of global optimality, given an upper bound on the number of transitions along a hybrid trajectory. Not surprisingly, the optimal solution is hybrid in nature, in that it depends on not only the continuous control signals, but also on discrete decisions as to what domains the system's continuous state should go through in the first place.

The main benefit with the proposed approach lies in the fact that a hierarchical Dynamic Programming algorithm can be used to representing both a theoretical characterization of the hybrid solution's structural composition and, from a more application-driven point of view, a numerically implementable calculation rule

yielding to globally optimal solutions in a regional dynamics framework. The operation of the recursive algorithm is highlighted by the consideration of numerous examples, among them, a heterogeneous multi-agent problem.

CHAPTER I

INTRODUCTION AND OBJECTIVE

This chapter highlights the notable relevance of hybrid models in the characterization and description of complex dynamical systems, which have recently gained remarkable significance through the more and more advanced investigations in almost every modern field of application. In this context, the particular optimal control problem addressed in this work is presented, which is followed by a brief outline of the main contents in this thesis.

1.1 Motivation

During the last decade, a vast body of research on hybrid control systems has been produced, drawing its relevance from the fact that hybrid models are becoming more and more common. This trend is driven by the fact that many modern application domains involve complex systems, in which sub-system interconnections, mode-transitions, and heterogeneous computational devices are present.

Hybrid models, in which continuous and discrete dynamical components interact, have proved useful for capturing these types of phenomena. In fact, one can argue that the "hybridization" of the models is useful for two distinctly different reasons. The first reason is that by decomposing complex control tasks down into simpler building-blocks, the overall control design problem becomes easier, and as a result, a non-hybrid problem becomes hybrid by choice. The benefit from this divide and conquer approach is that a number of relatively easier conrol tasks can be solved, and the resulting controllers are then concatenated together to produce the global system behavior. As an example of this, consider the behavior-based robotics paradigm for mobile robot navigation. The other reason why hybrid models are useful is that some problems are inherently hybrid in that mode-transitions

occur in response to events, that may be exogeneous as well as being triggered by events in the continuous state space.

In this work, the latter of these types of systems is considered, namely hybrid systems in which transitions between different dynamical regimes occur as the continuous state of the system reaches a given switching surface. In particular, the attention is focused on the optimal control problem associated with such systems. More precisely, given a specific cost function the goal is to determine the optimal path of going from a given initial state ξ_0 to a fixed final state ξ_T during a time horizon T, where T is also specified a priori. Note that stochastic approaches to similar problems were proposed in [27, 44].

Optimal control of hybrid systems is certainly not a new topic. For example, the hybrid maximum principle has been well-studied, and the community now has a clear grasp of what constitutes necessary optimality conditions for very general classes of hybrid systems [15, 60–62, 64]. Moreover, a number of results of a more computational flavor have complemented the work on the maximum principle, in which specialized classes of systems are considered. See for example [6, 23, 28, 54, 69]. These computational contributions typically fall in one of two camps, namely the camp in which the switching times are available to the controller as a design parameter [23, 69], or, the camp in which a more restrictive class of model dynamics, for example, piecewise linear or affine discrete-time models, is considered, for which mixed-integer programming techniques can be used [6, 28].

The novelty of the solution herein lies in the treatment of global optimality conditions for the general class of regional dynamics systems, where the hybrid nature is inherent in that transitions between different dynamical regimes are triggered as the continuous state intersects certain surfaces in the state space. Based on a hierarchical structuring of the optimal control problem associated with such systems, along the lines of [11, 12, 32], a Hybrid Bellman Equation is derived, representing both, a mathematical formulation and theoretical characterization of the hybrid solution's structural composition and, from a more application-driven

point of view, an implementable, numerically computable calculation rule.

1.2 Outline of the Thesis

This work is structured as follows:

In Chapter 2, a brief summary on optimal control theory is given, particularly focusing on results, definitions, and ideas fundamental for the understanding of our subsequent discussions on hybrid systems. Chapter 3 presents an explanatory and illustrative description of the hybrid optimal control problem addressed in this work and, moreover, provides a rough sketch of the proposed solution scheme. The purpose of Chapter 4 is to present a first, preliminary approach to the mathematically precise treatment of the general hybrid point-to-point problem introduced in Chapter 3 considering only two regions, i.e., two different discrete modes, and a simplified dynamic behavior. The *bimodal* system (with limited dynamics) is considered at first with the intention of illustrating various important mathematical assumptions and definitions needed later to accurately formulate and solve a *general* hybrid point-to-point problem. Finally, in Chapter 5, global optimality conditions for the multiregional point-to-point problem are presented and the main theorem of this work, the Hybrid Bellman Equation for multimodal systems, is stated. However, Chapter 6 shows the universal validity of the derived automaton-based Dynamic Programming recursion, which applies, in fact, to a much huger class of hybrid optimal control problems characterized by more general dynamic regimes, differently defined geometric structures, and special cost functions. In the end, in Chapter 7, the powerful computational operation of the proposed approach, established in Chapter 4 and Chapter 5, is proven by solving a hybrid multi-agent problem. Chapter 8 concludes the investigations in this thesis and highlights the work's main contributions.

CHAPTER II

BACKGROUND – OPTIMAL CONTROL

This chapter provides a summary on optimal control theory and is particularly focused on results, definitions, and ideas fundamental for the understanding of our subsequent discussions.

Even advanced readers are highly encouraged to read the following sections not only because they present a concise review of two basic optimal control approaches, the Dynamic Programing algorithm (Section 2.2) and Pontryagin's Minimum Principle (Section 2.3), but especially because, in the course of this work, frequent references are made to notations, conclusions and considerations highlighted in this chapter. In fact, many concepts, as for example the Principle of Optimality and the idea of defining a cost-to-go function, can be adapted to the subsequently arising problems and are useful means to their solution. However, in order to obtain a profound introduction to optimal control and to understand the theoretical and mathematical background, the reader is referred to a recommendable selection of books presented in Section 2.4.

Now, as a first step, a formal mathematical definition of the optimal control problem is given.

2.1 The Optimal Control Problem

The goal in the theory of optimal control is to determine, given a dynamical system, the system input which minimizes a specific performance measure.

This section presents a formal statement of the optimal control problem in terms of a *continuous-time* formulation, Section 2.1.1, and the corresponding *discrete-time* statement, Section 2.1.2. The continuous-time version provides an excellent basis for the more advanced expressions necessary to define the hybrid

optimal control problem in Chapter 4 and Chapter 5. The discrete-time version is used to introduce the Dynamic Programming method in Section 2.2 which plays an important role in deriving the main result of this thesis, the Hybrid Bellman Equation for multi-modal systems.

2.1.1 The Continuous-Time Optimal Control Problem

First, consider a continuous-time system described by the equation

$$\dot{x}(t) = f(x(t), u(t), t)\,, \tag{2.1}$$

where u is the control input of the system and f is assumed to be continuously differentiable in x (for all u). Besides the given dynamics (2.1), which represent a mathematical model of the real system, further key elements specifying an optimal control problem are the performance measure J, which is to be minimized, and the constraints on the state x and input u of the system. The performance measure J, also referred to as the cost functional, is defined as the sum of an integral term, called stage cost, and a term representing the terminal penalty, namely,

$$J = \int_{t_0}^{t_f} \ell(x(t), u(t), t)\, dt + \Phi(x(t_f), t_f). \tag{2.2}$$

In this work, only finite time intervals $\mathcal{T} = [t_0, t_f]$ with $-\infty < t_0 \leq t_f < \infty$ are taken into account. Moreover, due to the time-invariance of the problems considered in the following, $\mathcal{T} = [0, T]\,,\ T < \infty$ can be chosen without loss of generality. The state constraints are given by

$$x(\tau) \in \mathcal{X} \subseteq \mathbb{R}^n, \quad \tau \in (t_0, t_f)\,. \tag{2.3}$$

Additionally, the starting point $x(t_0)$ and the end point $x(t_f)$ are required to satisfy

$$x(t_0) \in \mathcal{X}_0 \subseteq \overline{\mathcal{X}} \quad \text{and} \quad x(t_f) \in \mathcal{X}_f \subseteq \overline{\mathcal{X}}, \tag{2.4}$$

respectively, where $\overline{\mathcal{X}}$ denotes the closure of $\mathcal{X}$. One special case of (2.4), arising in the optimal control problem considered in this work, later also referred to as the hybrid point-to-point problem, is the definition of a fixed initial state $x(t_0) =$

ξ_0 and a fixed final state $x(t_f) = \xi_f$. As regards the input of the system, the constraints are represented by

$$u(\tau) \in \mathcal{U}(x,t) \subseteq \mathbb{R}^m, \quad \tau \in [t_0, t_f]\,. \tag{2.5}$$

However, for the sake of simplicity input constraints are neglected in the subsequent chapters of the thesis.

Now, by making use of the notation $g(\cdot)$ for a function g defined only on the interval $\mathcal{T} = [t_0, t_f]$, i.e. $g(\cdot) = \{g(\tau) : \tau \in [t_0, t_f]\}$, the general optimal control problem can be stated as follows:

Problem 2.1 (The continuous-time optimal control problem, cf. [42]). *Determine the input signal $u^*(\cdot)$, such that*

(i) the equations specifying the system dynamics (2.1)*, the state constraints* (2.3)*, the boundary conditions* (2.4) *and the input constraints* (2.5) *are satisfied and*

(ii) the cost functional J given by (2.2) *takes on the least possible value.*

That is to say,

$$\inf_{u(\cdot)} J \quad \textit{subject to} \quad \textit{the equations } (2.1) - (2.5). \tag{2.6}$$

□

The input $u^*(\cdot)$ is called an *optimal* control input. The corresponding cost functional $J^*(u^*(\cdot))$, referred to as value function, satisfies $J^*(u^*(\cdot)) \leq J(u(\cdot))$ for all *admissible* $u(\cdot)$, i.e. for all input signals $u(\cdot)$ meeting the specifications (2.1) – (2.5). Figure 2.1 gives an illustrative example of the presented optimal control setting. The resulting optimal trajectory $x^*(\cdot)$ associated with the optimal input signal $u^*(\cdot)$ is depicted for the case of an one-dimensional state $x(t) \in \mathbb{R}$.

2.1.2 The Discrete-Time Optimal Control Problem

The discrete-time optimal control problem can be developed in an analogous way. In this case, only a finite number of input values, namely an *optimal* input sequence

$\{u^*(k)\} = \{u^*(k_0), \ldots, u^*(k_f - 1)\}$, has to be determined. The following is a formal, mathematical way of expressing this optimal control problem:

Problem 2.2 (The discrete-time optimal control problem, cf. [9]).

$$\inf_{\{u(k)\}} J, \quad \textit{where} \quad J = \sum_{k=k_0}^{k_f-1} \ell\left(x(k), u(k), k\right) + \Phi\left(x(k_f), k_f\right), \tag{2.7}$$

subject to

$$\textit{the system dynamics} \qquad x(k+1) = f(x(k), u(k), k), \tag{2.8}$$

$$\textit{the state constraints} \qquad x(k) \in \mathcal{X}, \tag{2.9}$$

$$\textit{the boundary conditions} \qquad x(k_0) \in \mathcal{X}_0\,,\ x(k_f) \in \mathcal{X}_f, \qquad \textit{and} \tag{2.10}$$

$$\textit{the input constraints} \qquad u(k) \in \mathcal{U}(x(k), k), \tag{2.11}$$

where $k \in \{k_0, \ldots, k_f - 1\} \subseteq \mathbb{N} \cup \{0\}$ *and* $\infty \notin \mathbb{N}$. □

The concept of admissibility, introduced for continuous-time systems, can also be adopted in the case of a discrete-time optimal control problem. Additionally, a set of admissible input values $\mathcal{U}_{ad}(x(k), k)$ can be explicitly determined for each given value $x(k) \in \mathcal{X},\ k \in \{k_0, \ldots, k_f - 2\}$ using the equations (2.11), (2.8), and (2.9)

$$\mathcal{U}_{ad}(x(k), k) = \left\{u(k) \in \mathcal{U}(x(k), k) \;\middle|\; f(x(k), u(k), k) \in \mathcal{X}\right\}. \tag{2.12}$$

Furthermore, in order to take the final constraint $x(k_f) \in \mathcal{X}_f$ into account, the admissible set of input values $\mathcal{U}_{ad}$ for $k = k_f - 1$ and $x(k_f - 1) \in \mathcal{X}$ is defined as

$$\mathcal{U}_{ad}(x(k_f-1), k_f-1) = \left\{\upsilon \in \mathcal{U}(x(k_f - 1), k_f - 1) \;\middle|\; f(x(k_f - 1), \upsilon, k_f - 1) \in \mathcal{X}_f\right\}. \tag{2.13}$$

2.2 Dynamic Programming

A key concept of solving optimal control problems is Dynamic Programming (DP).

After having established the main objectives of optimal control in the previous section, a general idea of finding optimal solutions, namely the Principle of Optimality, is introduced (Section 2.2.1). Based on this principle, in Section 2.2.2, a

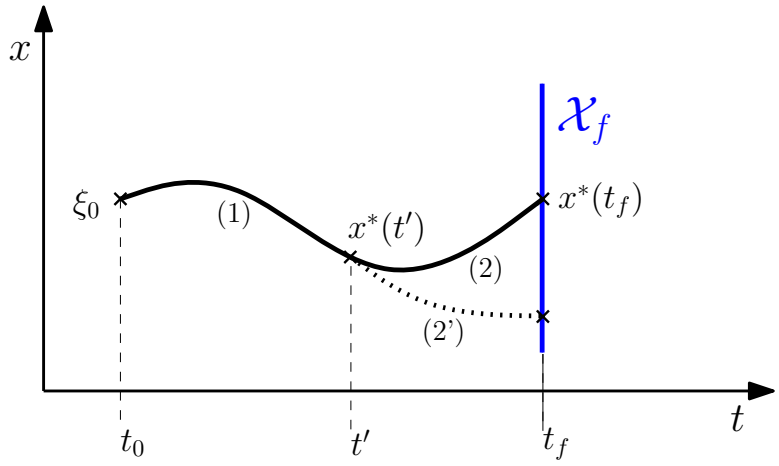


Figure 2.1: Solution of a Continuous-Time Optimal Control Problem with $x(t_0) = \xi_0 \in \mathbb{R},\ x(t_f) \in \mathcal{X}_f$ – The solid line represents the optimal trajectory.

Dynamic Programming (DP) algorithm is derived solving the discrete-time optimal control problem (Problem 2.2). However, this Dynamic Programming method represents an impressively wide-ranging concept valuable in solving very general *multistage decision problems* [55], that is, problems which can be divided into subproblems whose sequential solution can be used to find the solution of the overall problem. In particular, the hybrid optimal control problem considered in this work can be interpreted as a multistage decision problem and the ideas presented in this section are strongly involved in our hybrid optimal solution. The conclusion of this section discusses some crucial characteristics of the Dynamic Programming method. These properties hold also in the following hybrid optimal approach, which is presented in its most general form in Section 5.

2.2.1 The Principle of Optimality

The fundamental concept of Dynamic Programming lies in the Principle of Optimality, which can be stated as follows [4]:

An optimal policy has the property that whatever the initial state and the initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

Another more specific way of presenting this idea is shown in Figure 2.1; see

also [38]. The optimal state trajectory $x^*(\cdot)$ has the property that no matter how an intermediate state $x^*(t')$ is reached, the rest of the trajectory, designated path (2), should coincide with an optimal trajectory calculated with the intermediate point $x^*(t')$ as the starting point. If this is not the case, i.e. if a "more optimal" path (2') exists determined with the initial condition $x^*(t')$, it is possible to concatenate the first part (1), from $x^*(t_0)$ to $x^*(t')$, with path (2') resulting in a smaller overall cost which contradicts the supposed optimality of (1)–(2). Simply said, end pieces of optimal trajectories $(u^*(\cdot), x^*(\cdot))$ are optimal.

In fact, the following hybrid optimal control approach and especially the ideas in Section 3.2 are a direct consequence of this intuitive principle.

2.2.2 Backwards Dynamic Programming Recursion

In the following, the Principle of Optimality is used to derive a strategy for solving the discrete-time optimal control problem of Section 2.1 leading to a recurrence relation known as the *Bellman equation*. Later, in Section 4.4 and Section 5.6 a *Hybrid* Bellman Equation is developed which is closely related to the result shown below. As a consequence, the solution techniques as well as the computational considerations presented in this section can be transfered directly to the hybrid control problem.

An important step in solving the discrete-time optimal control problem, introduced in Section 2.1, is the definition of a so-called *cost-to-go* function V. Informally stated, $V(x(m), m)$ is the smallest cost of going from state $x(m) \in \mathcal{X}$ at stage m to a terminal state $x(k_f) \in \mathcal{X}_f$ while following the dynamics (2.8) and satisfying the input and state constraints, (2.11) and (2.9), respectively. Recalling the definitions (2.12) and (2.13), the cost-to-go function $V(x(m), m)$ is precisely

given by

$$V(x(m),m) = \inf_{\{u(m),\ldots,u(k_f-1)\}} \left\{ \sum_{k=m}^{k_f-1} \ell\left(x(k),u(k),k\right) + \Phi\left(x(k_f),k_f\right) \right\} \tag{2.14}$$

subject to

$$\begin{aligned} &x(k+1) = f(x(k),u(k),k) \\ &u(k) \in \mathcal{U}_{ad}(x(k),k) \end{aligned} \qquad \forall\, k \in \{m,\,\ldots,\,k_f-1\},$$

where $x(m) \in \mathcal{X}$ and $k_0 \leq m \leq k_f - 1$. Using the above definition along with the Principle of Optimality, a recurrence relationship can be derived providing a solution for the considered optimal control problem.

Theorem 2.1 (Backwards Dynamic Programming Recursion, cf. [9]). *The cost-to-go function V, defined by* (2.14)*, satisfies the following recursive equation*

$$V(x(m),m) = \inf_{u\in\mathcal{U}_{ad}(m,\,x(m))} \Big\{ \ell\big(x(m),u,m\big) + V\big(f(x(m),u,m),\,m+1\big) \Big\}, \tag{2.15}$$

where $m = k_0,\,\ldots,\,k_f-1$ and $x(m) \in \mathcal{X}$. By performing a backwards recursion starting with $V(x(k_f),k_f) = \Phi\left(x(k_f),k_f\right)$ and evaluating equation (2.15) *successively for $m = k_f-1,\,\ldots,\,k_0$, the cost-to-go functions $V(x(m),m)$ are found for all $x(m) \in \mathcal{X}$, $m = k_f-1,\,\ldots,\,k_0$. As a result, the desired value function J^* associated with Problem 2.2 can be given by*

$$J^* = \inf_{x(k_0)\in\mathcal{X}_0} \Big\{ V\big(x(k_0),k_0\big) \Big\}. \tag{2.16}$$

The optimal input values $u^*(m)$, $m = k_0,\,\ldots,\,k_f-1$ are obtained immediately while calculating the cost-to-go functions $V(x(m),m)$:

$$u^*(m) = \arg\inf_{u\in\mathcal{U}_{ad}(m,x(m))} \Big\{ \ell\big(x(m),u,m\big) + V\big(f(x(m),u,m),\,m+1\big) \Big\}. \tag{2.17}$$

□

The *Bellman equation* (2.15) makes it possible to calculate V successively beginning from $V(x(k_f),k_f) = \Phi\left(x(k_f),k_f\right)$ and continuing to the functions $V(x(k_0),k_0)$

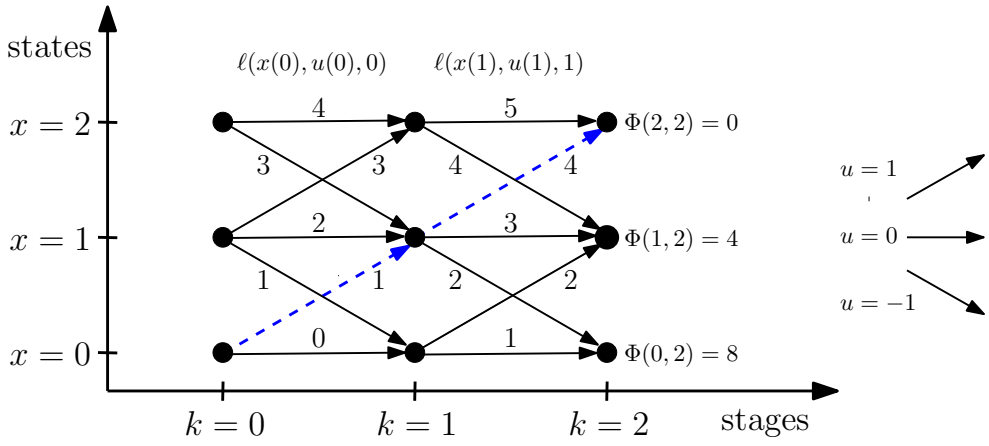


Figure 2.2: Example Illustrating the Dynamic Programming Method – The solid arrows represent possible transitions between states. The dashed arrows display the optimal path corresponding to the specifications given in Example 2.1.

needed to get the desired value function J^*. The technique itself used to find J^* with the aid of (2.15) is called dynamic programming [9].

In the following, the beauty of the Dynamic Programming approach is illustrated by solving a simple discrete-time optimal control problem.

Example 2.1. Shown in Figure 2.2 is a controlled discrete-time system with $x(k+1) = x(k) + u(k)$, $\mathcal{X} = \{0, 1, 2\}$, $k_0 = 0$ and $k_f = 2$. A cost function J, as defined in equation (2.7), is given through

$$\ell(x(k), u(k), k) = 2x(k) + u(k) + k \quad \text{and} \quad \Phi(x(k_f), k_f) = 8 - 4x(k_f),$$

where the values of ℓ are displayed in Figure 2.2 for each possible state transition illustrated by an arrow connecting the corresponding states. Furthermore, the arrows indicate all possible choices of input values at a given state $x(k)$, i.e. they specify the set $\mathcal{U}(x(k), k)$. The boundary conditions are given by $\mathcal{X}_0 = \mathcal{X}_f = \mathcal{X}$. Now, using (2.15) and working backwards, the "first" cost-to-go functions to be determined are $V(x(k), k)$, $x(k) \in \mathcal{X}$ with $k = k_f - 1 = 1$. Examplarily, the cost-to-go function $V(2, 1)$ is calculated

$$V(2,1) = \inf_{u \in \{0,-1\}} \{\ell(x(2,u,1) + \Phi(f(2,u,1),\, 2)\} = \inf\{5+0,\, 4+4\} = 5,$$

where in this case $u = 0$ leads to the smallest value for $V(2, 1)$. Analogously, $V(1, 1) = 4$ and $V(0, 1) = 6$ can be computed. After having performed this first

minimization, each "node" $x(1) \in \mathcal{X}$ is associated with an optimal cost-to-go value $V(x(1), 1)$ and a corresponding optimal path, leading from $x(1)$ to the final set $\mathcal{X}_f$. Important to note is that $V(x(1), 1)$ must be calculated for every value $x(1) \in \mathcal{X}$ in order to complete the next recursion step, the determination of $V(x(0), 0)$. As an example, $V(0, 0)$ is determined

$$V(0,0) = \inf_{u \in \{0,1\}} \{\ell(0,u,0) + V(f(0,u,0),1)\} = \inf\{0+6,\ 1+4\} = 5.$$

Simultaneously, the minimizing input value is obtained, in this case $u = 0$. In a similar way, the values $V(1, 0) = 6$ and $V(2, 0) = 7$ are computed. Note that instead of accomplishing the minimization as a whole problem, i.e. determining the cost functions for all 17 allowed paths and minimizing over all these possibilities, a sequence of very simple minimization problems is solved which involve comparing only two or three numbers at each node. Finally, by applying equation (2.16) the value function J^* can be computed

$$J^* = \inf\{V(0,0),\ V(1,0),\ V(2,0)\} = \inf\{5,\ 6,\ 7\} = 5.$$

The dashed arrows in Figure 2.2 illustrate the optimal path and picture the optimal input sequence $\{u^*(k)\} = \{1, 1\}$. □

References [4, 5, 9, 13, 38] describe further applications of the Dynamic Programming algorithm and a wider range of illustrative examples.

2.2.3 Characteristics and Conclusion

The conclusion of this section on Dynamic Programming identifies and describes important properties in view of the later application to hybrid systems.

As highlighted in the previous example, in order to solve the original optimal control problem, an entire set of minimization problems is solved. Precisely, all cost-to-go functions $V(x(k), k),\ x(k) \in \mathcal{X},\ k_0 \leq k \leq k_f$ are computed, starting with $k = k_f$ and proceeding in a backwards manner. As a result, we obtain not only the "quality" $V(x(k), k)$ of each node $x(k)$ in terms of its future performance under an optimal policy but also the appropriate optimal control input itself (2.17), and,

with this, the associated optimal path from $x(k)$ to $\mathcal{X}_f$. In fact, this procedure is equivalent to finding concurrently the optimal trajectories corresponding to various initial sets $\mathcal{X}_0$ at different stages $k_{0,new}$: $k_0 \leq k_{0,new} \leq k_f$. Therefore, solving the optimal control problem under changed initial conditions requires very little additional calculation.

In order to consider some further computational aspects, the total number of additions n_{add} and the total number of comparisons n_{comp} is chosen as a measure for the inherent computational effort needed to complete a DP algorithm. For the number of comparisons, it is supposed that only two members can be compared at a time. In the following, an optimal control problem with $|\mathcal{U}(x(k), k)| = a$ different input values, $|\mathcal{X}| = b$ state values, and $|\{k_0, \ldots, k_f\}| = c$ time stages is regarded, where $|\mathcal{S}|$ denotes the number of elements in the set $\mathcal{S}$. For simplicity, $\mathcal{X}_0 = \mathcal{X}_f = \mathcal{X}$ is chosen. When this problem is solved by using the introduced Dynamic Programming algorithm, i.e. equations (2.15) and (2.16), the total number of additions is generally given by

$$n_{\text{add}} = ab(c-1) \tag{2.18}$$

and the total number of comparisons adds up to

$$n_{\text{comp}} = ab(c-1) + b - 1. \tag{2.19}$$

Consequently, increasing one of the numbers a, b, c and fixing the other two, the computational effort increases *linearly*. To demonstrate the quality and importance of this result, the Dynamic Programming approach is compared with the tedious way of calculating the cost for *all* possible routes from $\mathcal{X}_0$ to $\mathcal{X}_f$ and minimizing afterwards. Since the number of possible routes n_{paths} from $\mathcal{X}_0$ to $\mathcal{X}_f$ runs up to $n_{\text{paths}} = a^{c-1}b$, the computational effort of this method increases polynomially while increasing the number of input values a and even increases exponentially if the number of time stages c grows. Therefore, from this point of view, the Dynamic Programming method is highly efficient and reduces the number of calculations dramatically, see also [1]. However, assumed the discrete states

are obtained by quantizing a continuous state space $\mathcal{X} \subseteq \mathbb{R}^n$, the number b of states $x(k)$ increases exponentially if the dimensionality n of the continuous state space raises. Therefore, in practice, for high-dimensional systems the number of high-speed storage locations becomes restrictive [1]. Being forced to use low-speed storage, the solution of the optimal control problem may become time-consuming. This difficulty, the so-called "curse of dimensionality" [4], is observed especially while solving the multi-agent problem in Chapter 7.

Finally, especially with respect to the later approach to multimodal systems, it is valuable again to emphasize the universal validity of the Principle of Optimality and the general applicability of the Dynamic Programming concept. In the following chapters, the idea of defining a cost-to-go function and deriving a recurrence relationship is adopted and expanded in a sophisticated way. Used in this approach and of great importance either way is the property that, in the context of a *decision problem* [55], the optimization technique chosen to solve the individual subproblems does not affect the Dynamic Programming technique itself by any means.

Besides the Dynamic Programming method which provides a globally optimal solution, Theorem 2.1, and which is mainly used in a discrete framework, this work is based on another concept solving optimal control problems which is introduced in the following section.

2.3 Pontryagin's Minimum Principle

Pontryagin's Minimum Principle (PMP) represents a basic optimal control approach primarily dealing with continuous-time problems.

The goal of this section is to provide the reader with the main result, known as Pontryagin's Minimum Principle (PMP) presented as an adaptation to the problems arising in the subsequent chapters. In Chapter 4, Chapter 5, and also in the application-driven investigations in Chapter 7, the result serves as a method to solve the individual subproblems of our *multistage decision problem* [55] and is,

therefore, a crucial element in the solution of the entire hybrid control problem. In the course of this section, the solution to these subproblems, in case of a linear system and quadratic cost functional, is shown as an example for applying Pontryagin's Minimum Principle to continuous-time optimal control problems.

2.3.1 The Minimum Principle

This section considers a continuous-time optimal control setup as in Problem 2.1, however, with the following specifications: A time-invariant framework is chosen, i.e.

$$f = f\left(x(t), u(t)\right), \quad \ell = \ell\left(x(t), u(t)\right), \quad \Phi = \Phi(x(t_f)), \tag{2.20}$$

and, cf. the comments on equation (2.2),

$$\mathcal{T} = [t_0, t_f] = [0, T]\,. \tag{2.21}$$

The restrictions on the state values are given by

$$\mathcal{X} = \mathbb{R}^n, \quad \mathcal{X}_0 = \{x \in \mathcal{X} \mid h_0(x) = 0\}, \quad \text{and} \quad \mathcal{X}_f = \{x \in \mathcal{X} \mid h_f(x) = 0\}, \tag{2.22}$$

where $h_0(x) \in \mathbb{R}^p$ and $h_f(x) \in \mathbb{R}^q$. No constraints are enforced on the input values and therefore

$$\mathcal{U}(x, t) = \mathbb{R}^m. \tag{2.23}$$

Additionally, the following assumptions hold:

Assumption 2.1. *The functions $\ell\left(x, u\right)$ and $f\left(x, u\right)$ are continuous and continuously differentiable with respect to x and u. Furthermore, $\Phi(x)$, $h_0(x)$, and $h_f(x)$ are continuous and continuously differentiable with respect to x.* □

Now, in order to state Pontryagin's optimality conditions concisely, some useful notations are introduced first. The absolute value of a number $a \in \mathbb{R}$ is denoted by $|a|$, whereas $\|A\|$ represents the norm of a vector or a matrix $A \in \mathbb{R}^{n \times m},\ n, m \in \mathbb{N}$. The gradient of a scalar function $F(x) \in \mathbb{R}$, $x \in \mathbb{R}^n$ is defined as

$$\frac{\partial F}{\partial x}(x) = \begin{pmatrix} \dfrac{\partial F}{\partial x_1} & \dfrac{\partial F}{\partial x_2} & \cdots & \dfrac{\partial F}{\partial x_n} \end{pmatrix} \tag{2.24}$$

and the Jacobian of a vector-valued function $f(x) = (f_1(x) \;\; f_2(x) \;\; \cdots \;\; f_m(x))'$, $x \in \mathbb{R}^n$, where $(\cdot)'$ symbolizes the transpose of the vector, is given by

$$\frac{\partial f}{\partial x} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}. \tag{2.25}$$

Using these notations and the specifications above, Pontryagin's Minimum Principle can be formulated as follows:

Theorem 2.2 (Pontryagin's Minimum Principle, cf. [52]). *Let $(x^*(\cdot), u^*(\cdot))$ be an optimal solution to Problem 2.1, specified by Assumption 2.1 and equations (2.20)–(2.23). Then, there exist a $\lambda_0^* \in \mathbb{R}$,*

$$\mu_0^* = \left(\mu_{0,1}^*, \ldots, \mu_{0,p}^*\right) \in \mathbb{R}^{1\times p}, \quad \text{and} \quad \mu_f^* = \left(\mu_{f,1}^*, \ldots, \mu_{f,q}^*\right) \in \mathbb{R}^{1\times q}$$

where $|\lambda_0^*| + \|\mu_0^*\| + \left\|\mu_f^*\right\| \neq 0$ and $\lambda_0^* \leq 0$, such that for the Hamiltonian defined as

$$H\left(x(t), \psi(t), \lambda_0, u(t)\right) = \psi(t) f\left(x(t), u(t)\right) - \lambda_0 \ell\left(x(t), u(t)\right) \tag{2.26}$$

and the costate $\psi^*(\cdot)$, solving the adjoint equation

$$\begin{aligned} \dot{\psi}^*(t) &= -\frac{\partial H}{\partial x}\left(x^*(t), \psi^*(t), \lambda_0^*, u^*(t)\right) \\ &= -\psi^*(t)\frac{\partial f}{\partial x}\left(x^*(t), u^*(t)\right) + \lambda_0^*\frac{\partial \ell}{\partial x}\left(x^*(t), u^*(t)\right), \end{aligned} \tag{2.27}$$

the following conditions hold:

1. *the Minimum Principle*

$$\frac{\partial H}{\partial u}\left(x^*(t), \psi^*(t), \lambda_0^*, u^*(t)\right) = 0, \quad t \in \mathcal{T}, \tag{2.28}$$

2. *the transversality conditions for the initial costate*

$$\psi^*(0) = -\mu_0^*\frac{\partial h_0}{\partial x}\left(x^*(0)\right), \tag{2.29}$$

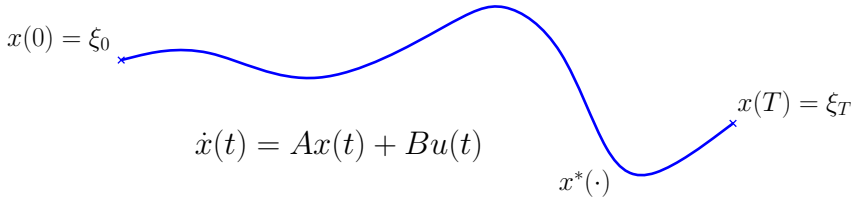


Figure 2.3: A Basic Optimal Control Problem: Finding the Optimal Path, starting at a given point ξ_0 and going to a fixed state ξ_T.

3. the transversality conditions for the final costate

$$\psi^*(T) = -\lambda_0^* \frac{\partial \Phi}{\partial x}\left(x^*(T)\right) + \mu_f^* \frac{\partial h_f}{\partial x}\left(x^*(T)\right). \tag{2.30}$$

□

This theorem, obtained by employing local variational methods, provides a first-order *necessary* optimality condition. Therefore, not every trajectory $(x(\cdot), u(\cdot))$ satisfying the equations (2.26)–(2.30) is optimal, however, any trajectory represents a possible candidate for an optimal trajectory.

The following example shows how the Minimum Principle can actually be used to determine optimal trajectories. Moreover, this example represents a key element in solving the subsequent hybrid optimal control problem in case of linear dynamics and quadratic cost functional. However, the point-to-point problem considered in the following has to be solved under a large number of different conditions in order to finally obtain the *hybrid* trajectory. Therefore, in the subsequent section, some computational issues are also taken into account.

2.3.2 The Point-To-Point Problem in One Region

Our goal is to find the *optimal path* for going from a given starting point $x(0) = \xi_0$ to a fixed final state $x(T) = \xi_T$ under a specific cost functional J, where the time horizon T is known a priori. For this basic optimal control problem, illustrated in Figure 2.3, a *linear dynamical behavior* and a *quadratic cost functional* are assumed. Precisely, this problem can be formulated in the previously introduced optimal control framework (cf. Problem 2.1 with equations (2.20)–(2.23) and Assumption 2.1) as described in the following:

- The system dynamics are given by

$$f\left(x(t), u(t)\right) = Ax(t) + Bu(t) \quad \text{with (A,B) controllable.} \tag{2.31}$$

- The cost functional is defined with

$$\ell\left(x(t), u(t)\right) = x'(t)\, Q\, x(t) + u'(t)\, R\, u(t) \quad \text{and} \quad \Phi(x(t_f)) = 0, \tag{2.32}$$

  where $Q = Q' \geq 0$ and $R = R' > 0$.

- The boundary conditions can be written as

$$h_0(x) = x - \xi_0 \quad \text{and} \quad h_f(x) = x - \xi_T. \tag{2.33}$$

The notation $A > 0$ is used to express the positive definiteness of a (symmetric) matrix A; $A \geq 0$ denotes a positive semi-definite (symmetric) matrix A..

As a consequence of Theorem 2.2, for an input u to be optimal it must satisfy the Minimum Principle (2.28). From this condition, a function $u = u(x, \psi)$ is obtained. In our case, with the choice of $\lambda_0 = -1$, the Hamiltonian $H(x, \psi, u)$ (2.26) is read as

$$H(x, \psi, u) = \psi\left(Ax + Bu\right) + x'Qx + u'Ru \tag{2.34}$$

and equation (2.28) provides

$$\frac{\partial H}{\partial u} \stackrel{!}{=} 0 \quad \Leftrightarrow \quad u(x, \psi) = -\frac{1}{2} R^{-1} B' \psi'. \tag{2.35}$$

In addition, along with the equations (2.27), (2.29), and (2.30), the dynamics of the costate ψ are given by

$$\dot{\psi} = -2x'Q - \psi A, \qquad \psi(0) = -\mu_0,\ \psi(T) = \mu_f, \tag{2.36}$$

where μ_0 and μ_f are arbitrary. The precedent considerations (2.35) and (2.36) together with the system dynamics $\dot{x} = Ax + Bu,\ x(0) = \xi_0,\ x(T) = \xi_T$ yield to a two-point boundary value problem (TPBVP), which is written as

$$\begin{pmatrix} \dot{x} \\ \dot{\psi}' \end{pmatrix} = \underbrace{\begin{pmatrix} A & -\frac{1}{2} B R^{-1} B' \\ -2Q & -A' \end{pmatrix}}_{M} \begin{pmatrix} x \\ \psi' \end{pmatrix}, \tag{2.37}$$

with the boundary conditions $x(0) = \xi_0,\ x(T) = \xi_T$. Here, $(\cdot)^{-1}$ denotes the inverse of a matrix. Note that the values $\psi(0),\ \psi(T)$ can be chosen arbitrarily.

Actually, with the controllability assumption (2.31), it can be shown, cf. [3, 39, 43] that our particular optimal control problem has a unique optimal solution, which, of course, necessarily has to satisfy (2.37). Consequently, the optimal trajectory is determined as

$$\begin{pmatrix} x^*(t) \\ \psi^*(t)' \end{pmatrix} = e^{Mt} \begin{pmatrix} \xi_0 \\ \psi^*(0)' \end{pmatrix}, \tag{2.38}$$

where the value $\psi^*(0)'$ is to be chosen in a such way that the boundary conditions on x are satisfied, i.e. with $e^{Mt} = \left(\begin{smallmatrix} \phi_{11}(t) & \phi_{12}(t) \\ \phi_{21}(t) & \phi_{22}(t) \end{smallmatrix}\right)$

$$\psi^*(0)' = \phi_{11}^{-1}(T)\left(\xi_T - \phi_{11}(T)\,\xi_0\right). \tag{2.39}$$

This result, combined with equations (2.38) and (2.35), provides the solution to the point-to-point optimal control problem.

As mentioned before, the point-to-point problem emerges while dealing with hybrid optimal control problems, as can be seen in the following chapters. Actually, the presented approach solves the entirety of subproblems arising in our considerations on multimodal systems. As can be seen in the further endeavour, the optimal hybrid trajectories are a concatenation of the optimal segments calculated in this section. Already at this point, it is important to highlight that in order to decide how to put together the several parts $(x^*(\cdot), u^*(\cdot))$, provided by the calculations above, together, the value functions J^* associated with each optimal section $(x^*(\cdot), u^*(\cdot))$ are indispensable. Of course, for an optimal part $(x^*(\cdot), u^*(\cdot))$, the value J^* can be obtained by using the integral equation (2.2). However, from a computational point of view, the numerical integration necessary to calculate J^* for each of the parts is very time-consuming. Therefore, a more favorable way of determining the value function J^* is derived.

Proposition 2.1 (The Value Function). *The value function J^* associated with the optimal trajectory $(x^*(\cdot), u^*(\cdot))$, defined by* (2.39)*,* (2.37)*, and* (2.35)*, is given*

by

$$J^* = \begin{pmatrix} \xi_0' & \psi^*(0) \end{pmatrix} \left[\left(e^{MT}\right)' S\, e^{MT} - S \right] \begin{pmatrix} \xi_0 \\ \psi^*(0)' \end{pmatrix}, \tag{2.40}$$

where the equations (2.37) *and* (2.39) *specify* M *and* $\psi^*(0)$*, respectively, and the matrix* S *is defined as*

$$S = -\frac{1}{4} \begin{pmatrix} 0 & I_{n\times n} \\ I_{n\times n} & 0 \end{pmatrix} \tag{2.41}$$

with $I_{n\times n} \in \mathbb{R}^{n\times n}$ *denoting the identity matrix.* □

Proof. Evaluating the cost function (2.2), specified by (2.32), for the optimal trajectory $(x^*(\cdot), u^*(\cdot))$, determined by (2.38) and (2.35), provides the following expression

$$J^* = \begin{pmatrix} \xi_0' & \psi^*(0) \end{pmatrix} \int_0^T \left(e^{Mt}\right)' \underbrace{\begin{pmatrix} Q & 0 \\ 0 & \frac{1}{4} B R^{-1} B' \end{pmatrix}}_{X} e^{Mt}\, dt \begin{pmatrix} \xi_0 \\ \psi^*(0)' \end{pmatrix}. \tag{2.42}$$

Now, the central idea of this proof is to find a matrix S satisfying the relation

$$\left(e^{Mt}\right)' X e^{Mt} = \frac{d}{dt}\left[\left(e^{Mt}\right)' S e^{Mt}\right] = \left(e^{Mt}\right)' \left[M'S + SM\right] e^{Mt}. \tag{2.43}$$

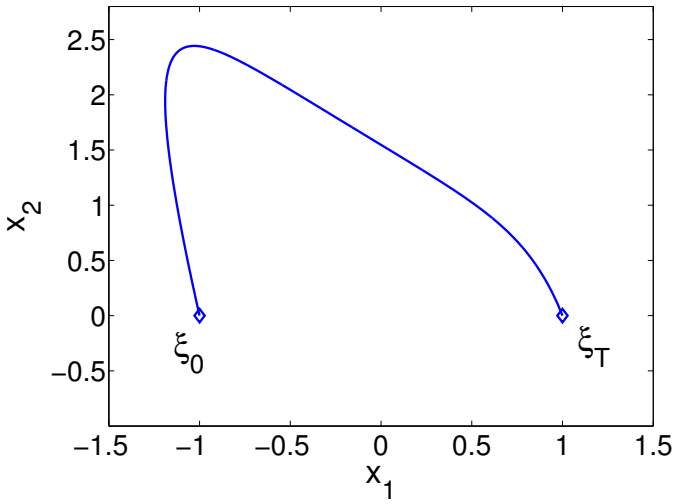
Supposed a matrix

$$S \in \mathbb{R}^{2n\times 2n} :\ M'S + SM = X \tag{2.44}$$

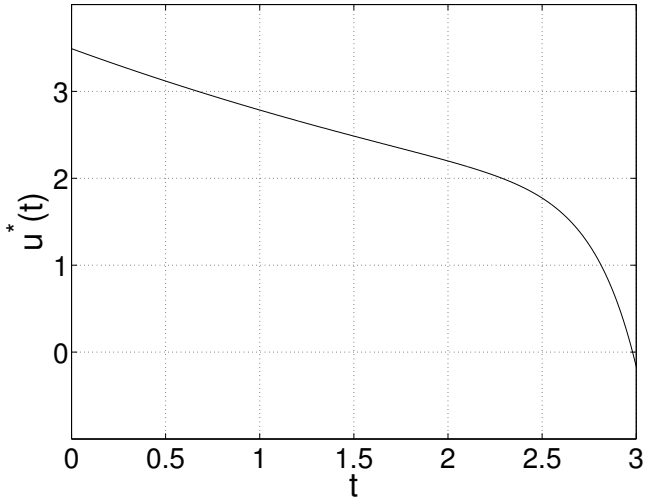
can be found, then, equation (2.40) follows immediately from (2.43) and (2.42). Indeed, with the choice for S, suggested in Proposition 2.1, and the matrix M, as defined in (2.37), equation (2.43) is satisfied and the proof is completed. ■

This result reveals an elegant way of computing the value function J^* for our considered point-to-point problem, especially when recalling the MATLAB function `expm(X)` which generates the matrix exponential e^X of an arbitrary matrix X.

The following numerical example presents an illustrative summary of the results in this section.



(a) Optimal States $(x_1^*(\cdot),\, x_2^*(\cdot))$



(b) Optimal Input $u^*(\cdot)$

Figure 2.4: Optimal Trajectory Corresponding To Example 2.2

Example 2.2. A planar system $x(t) = \begin{pmatrix} x_1(t) & x_2(t) \end{pmatrix}' \in \mathbb{R}^2$ is driven between the boundary points $\xi_0 = (-1\ 0)'$ and $\xi_T = (1\ 0)'$ under the system dynamics

$$\dot{x}(t) = \begin{pmatrix} 0.8 & 1 \\ -3 & 5 \end{pmatrix} x(t) + \begin{pmatrix} 0.3 \\ 3 \end{pmatrix} u(t) \tag{2.45}$$

during a time horizon $\mathcal{T} = [0, 3]$. The particular cost function (2.32) under consideration is the energy of the control signal $u(\cdot)$, i.e. $Q = 0$ and $R = 1$. Figure 2.4 shows the resulting optimal input $u^*(\cdot)$ and the corresponding state trajectory $(x_1^*(\cdot),\, x_2^*(\cdot))$, where the associated cost (2.40) takes the value $J^* \approx 18.82$. □

Finally, the introduced point-to-point problem not only solves the subproblems arising in the subsequent multimodal approach, but also represents a *special case* of the problem, where only one region is considered while generally the state space is divided into multiple regions (modes).

2.4 Further Reading

This section completes our introductory notes on optimal control by providing some references recommended for further studies.

The purpose of the preceding sections is to introduce and explain selected optimal control topics fundamental for the understanding of our later considerations on hybrid systems. While reviewing some basic optimal control methods, numerous indications are provided regarding the reasons that the concepts are presented and the way in which they are used in our following approaches. Examples closely related to the problems arising in the subsequent chapters are shown.

However, a thorough introduction to optimal control and the associated mathematical theory is explained in Kirk's well-known book [37] which covers a broad range of topics such as the Principle of Optimality, Dynamic Programming, calculus of variations, and Pontryagin's Minimum Principle. Furthermore, the continuous-time version of the Dynamic Programming method, the so-called Hamilton-Jacobi-Bellman equation, is derived. Another widely read book is the work of Bryson and Ho [13] which additionally considers a huge number of problems from various disciplines. A concise treatment of the optimal control basics is given by Knowles [38] and Locatelli [42]. Anderson and Moore [1] as well as Chui and Chen [18] especially dealing with linear systems.

Particularly focused on Dynamic Programming is the primary work of Bellman [4, 5], which also discusses a huge variety of different examples and applications. Pontryagin presents his results in [52]. A detailed and illustrative proof of Pontryagin's Principle is provided by Pinch [51].

CHAPTER III

THE HYBRID POINT-TO-POINT PROBLEM

The goal of this chapter is to provide an explanatory description of the hybrid framework considered in this work. After an illustrative introduction to the optimal control problem of interest in Section 3.1, Section 3.2 provides a rough sketch of the proposed solution scheme.

3.1 The Point-To-Point-Problem with Regional Dynamics

The optimal control problem of interest can be qualified as a *hybrid* point-to-point problem, where the governing dynamics $\dot{x} = f_i(x, u)$, $i \in \mathbb{N}$ vary depending on the region D_i, the continuous state x is evolving in.

This section introduces an overview of the central issue of our subsequent discussions, namely, a hybrid optimal control problem (HOCP) with regional dynamics. On the basis of an explicatory illustration, Figure 3.1, only the main features of this problem are described; however, the mathematical details are presented in Chapters 4 and 5.

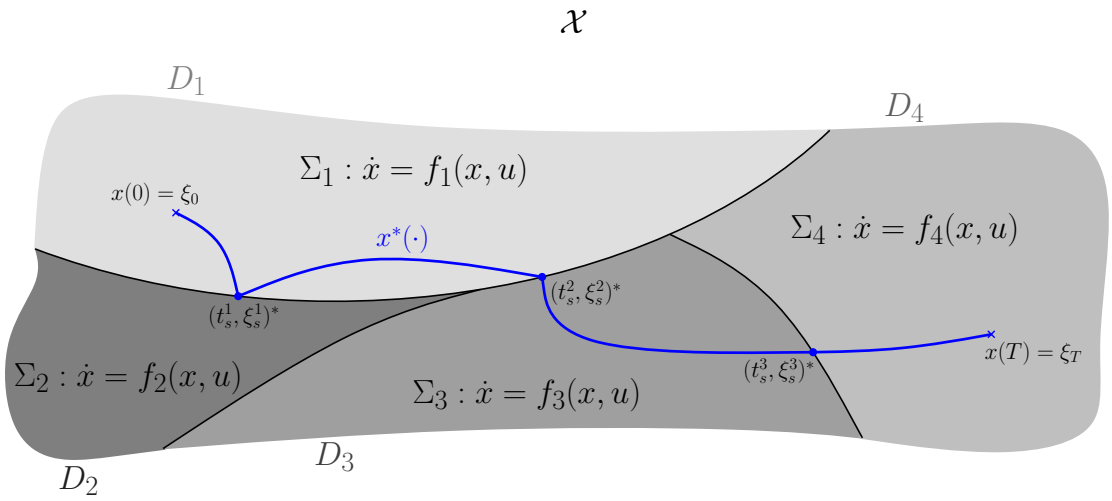


Figure 3.1: System with Regional Dynamics – Optimal Trajectory

As illustrated in Figure 3.1 by means of different shades of gray, the key characteristic in our hybrid control framework is a state space $\mathcal{X}$ partitioned into multiple regions $D_i \subseteq \mathcal{X}$, $i \in \mathbb{N}$. The black lines in Figure 3.1 depict the boundaries between each of two regions. Additionally, a particular dynamical system $\Sigma_i : \dot{x} = f_i(x, u)$, $i \in \mathbb{N}$ is associated with each region. Hence, the evolution of the continuous state x at a time t is determined by the vector field f_i corresponding to the state's current region $D_i : x(t) \in D_i$. The different regions can, consequently, be regarded as different *modes* of operation between which the system's behavior switches. As a result, besides the *continuous* evolution of the state $x(t) \in \mathcal{X}$ within a region, *discrete* transitions between different dynamical regimes occur as the continuous state of the system reaches specified boundaries. At a boundary, also referred to as a switching surface, two ways of further execution are possible:

(i) The trajectory "passes through" the switching surface and then evolves under another dynamical regime in a different region. This behavior is observed, for example, at the switching times $(t_s^2)^*$ and $(t_s^3)^*$ in Figure 3.1.

(ii) Another option for the continuous state's execution is to "bounce back" into the original region as it occurs at the switching point $(t_s^1, \xi_s^1)^*$, Figure 3.1. In this case, the dynamical regime remains the same; however, the control input may change discontinuously at that point.

Note that both ways of "transitioning", (i) as well as (ii), are called *switching* or *switch*. In summary, the preceding specifications describe a hybrid system, where mode-transitions are triggered by events in the continuous state space.

In fact, in this part of the research, the attention is focused particularly on the optimal control problem associated with such systems. More precisely, the goal is to solve the point-to-point problem, as considered in Section 2.3.2 for a linear system, in the case of the presented hybrid dynamics. Thus, given a cost function J, the goal is to determine the optimal path of moving from a given initial state $x(0) = \xi_0$ to a fixed final state $x(T) = \xi_T$ during an *a priori* specified

time horizon T, where the above regional dynamics are assumed. Indeed, this HOCP fits perfectly in the time-invariant optimal control setup previously defined by Problem 2.1 and equations (2.20)–(2.23), where the characteristic boundary conditions of the point-to-point problem are given by (2.33). Furthermore, the dynamical behavior in the hybrid control setup is described by the preceding explanations and illustrations (Figure 3.1). The blue trajectory in Figure 3.1 shows the optimal solution $x^*(\cdot)$ for one particular cost functional J. Note that in the following investigations, a "global" cost J is considered, which applies for each region D_i, $i \in \mathbb{N}$. However, among other generalizations, the subsequent approach also allows the definition of individual cost functions J_i for each region D_i, cf. Chapter 6.

The next section outlines the principle approach to solving the introduced point-to-point problem and can definitely be counted as one of the most important sections of this thesis.

3.2 The Dynamic Programming Approach

The novelty of the solution herein lies in the treatment of global optimality conditions for regional dynamics systems through a Dynamic Programming approach, where the hybrid point-to-point problem is considered as a *multistage decision problem* [55].

After having drawn an illustrative picture of the hybrid point-to-point problem (Section 3.1), the main focus of this section is on providing the central ideas of the solution procedure. Thereby, special emphasis is placed on an accessible presentation of the basic concepts helpful in understanding the detailed and precise approaches of Chapter 4 and Chapter 5, where mathematical notations and formalisms come to the fore.

The approach presented in this thesis is primarily based on the considerations in Section 2.2. Inspired by the Principle of Optimality and the Dynamic Programming method, important conclusions are made about the optimal solution of our

hybrid point-to-point problem. Having Figure 3.1 in mind or looking at Figure 3.2, a fundamental implication of the Principle of Optimality (Section 2.2.1) provides valuable insights into the evolution of the hybrid optimal solution $x^*(\cdot)$ depicted in blue:

The Principle of Optimality states implicitly that along an optimal hybrid trajectory $x^*(\cdot)$ the execution of the continuous state x between two consecutive switching points $(t_s^m, \xi_s^m)^*$, $(t_s^{m+1}, \xi_s^{m+1})^*$ is optimal and so is the part of the optimal hybrid input $u^*(\cdot)$ from the mth switching time to the $(m+1)$th switching time. Moreover, the region of the trajectory between $(t_s^m, \xi_s^m)^*$ and $(t_s^{m+1}, \xi_s^{m+1})^*$ must be an optimal location for this segment of the trajectory $x^*(\cdot)$.

The latter property is illustrated in Figure 3.2 for the section between $(t_s^2, \xi_s^2)^*$ and $(t_s^3, \xi_s^3)^*$, where both the optimal execution forced to stay in region 1 (solid line) and the trajectory optimally evolving in region 2 (dashed line) are depicted. In this example, region 1 represents the optimal location.

As a result of the considerations above, given two consecutive switching points $(t_s^m, \xi_s^m)^*$, $(t_s^{m+1}, \xi_s^{m+1})^*$, the optimal path between these points, the associated cost, and the optimal region of the corresponding segment can be determined without knowing more about the course of the hybrid optimal trajectory. Precisely speaking, in this case a standard (non-hybrid) state-constrained optimal control problem has to be solved [7, 17, 21, 22, 26, 30, 53]. The previous comments also apply to the initial and the final piece of the optimal hybrid trajectory, namely to the segment from $(0, \xi_0)$ to $(t_s^1, \xi_s^1)^*$ and to the path between $(t_s^{M^*}, \xi_s^{M^*})^*$ and (T, ξ_T), where M^* denotes the optimal number of switches associated with the hybrid optimal trajectory $x^*(\cdot)$ assuming a given upper bound $N \in \mathbb{N}$ on the total number of switches along a hybrid trajectory. The individual segments between optimal switching points $(t_s^m, \xi_s^m)^*$ as well as the initial and the final piece of the trajectory are depicted by solid blue curves in Figure 3.2 which concatenated result in the hybrid optimal trajectory $x^*(\cdot)$. Of course, the hybrid optimal *input* $u^*(\cdot)$ is also obtained by a concatenation of optimal (input) pieces calculated for

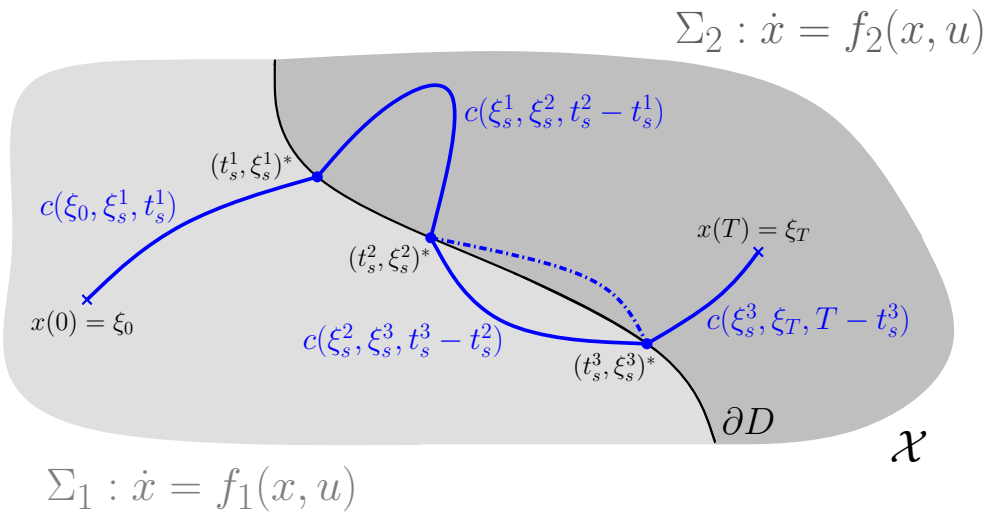


Figure 3.2: The Idea of Dynamic Programming In the Context of A Regional Dynamics System

each individual segment.

Recalling the time-invariance of the considered hybrid system (Section 3.1), an important step of our further proceeding is the introduction of the function $c(\xi_1, \xi_2, \Delta)$, $\xi_1, \xi_2 \in \mathcal{X}$ as the infimum of the costs associated with driving the hybrid system from the state ξ_1 to the point ξ_2 during a time horizon Δ *without a switch taking place*, i.e. while staying in one region with the points ξ_1, ξ_2 possibly on the boundary. Of course, this value can be computed between each pair $(\xi_1, \xi_2) \in \mathcal{X} \times \mathcal{X}$ and is infinity if the points ξ_1, ξ_2 lie in different regions. Figure 3.2 illustrates this idea by presenting the individual costs $c(\cdot, \cdot, \cdot)$ along the hybrid optimal trajectory. Obviously, the value function corresponding to the hybrid optimal trajectory in Figure 3.2 is now given by the sum of the displayed individual costs.

The previous observations highlight interesting attributes observed along optimal trajectories in regional dynamics systems. These properties are highly involved in the subsequent derivations of an algorithm to solve the hybrid point-to-point problem.

Primarily, the success of this approach is based on an appropriate division of the hybrid control problem into smaller subproblems. This goal is achieved by the definition of a specific cost-to-go function V^M, $M \in \mathbb{N} \cup \{0\}$ which implicitly

identifies the single subproblems. The idea of introducing a cost-to-go function as well as the consequent derivations, which finally (in Chapter 4 and 5) result in a Hybrid Bellman Equation (HBE), are closely related to the Dynamic Programming approach for discrete-time systems (Section 2.2.2). Discovering the analogies proves to be advantageous to gain further insight in the considered point-to-point problem. In the hybrid case, the cost-to-go function $V^M(\xi_1, \xi_2, \tau)$ is defined as the infimum of the costs of going from $\xi_1 \in \mathcal{X}$ to $\xi_2 \in \mathcal{X}$ during the time τ using *exactly M switches*. This central definition makes it possible to derive a recurrence relation solving the considered hybrid point-to-point problem. First, the optimal trajectory depicted in Figure 3.2 is used to explain the general procedure:

Example 3.1. Fundamental to this approach is the initial assumption of an a priori given number M of switches along the optimal hybrid trajectory. Consistent with the illustration in Figure 3.2, in this example $M = 3$ is chosen. Starting with the recursive procedure, our first subproblem is the determination of an optimal path of going from a point $\overline{\xi}$ on the boundary ∂D to the final point ξ_T *without any switch taking place* during a time horizon τ. Important to note is that this problem is solved for every point $\overline{\xi}$ on the boundary and every time $\tau \in [0, T]$ providing the values $V^0(\overline{\xi}, \xi_T, \tau) = c(\overline{\xi}, \xi_T, \tau)$ for each of the time-state combinations. Later, in order to being able to implement the proposed recursion method, the time interval $\mathcal{T} = [0, T]$ as well as the boundary itself are discretized resulting in a finite number of time-state pairs. Continuing the recursive scheme, the second subproblem determines the optimal path between a point $\overline{\xi} \in \partial D$ and the final point ξ_T using *exactly one switch*. The solution of this subproblem resulting in values $V^1(\overline{\xi}, \xi_T, \tau)$, where $\overline{\xi} \in \partial D$ and $\tau \in [0, T]$, is obtained by the equation

$$V^1(\overline{\xi}, \xi_T, \tau) = \inf_{\substack{\xi \in \partial D, \\ t \in [0,\tau]}} \Big\{ c(\overline{\xi}, \xi, t) + V^0(\xi, \xi_T, \tau - t) \Big\}, \tag{3.1}$$

which depends on the previously computed functions $V^0(\overline{\xi}, \xi_T, \tau)$ and, additionally, on the functions $c(\xi_1, \xi_2, t)$ with $\xi_1, \xi_2 \in \partial D$ and $t \in [0, T]$. The latter

functions can be calculated in advance, since they are not affected by the recursion equation (3.1) and are, in fact, completely independent of the given initial and final conditions, $x(0) = \xi_0$ and $x(T) = \xi_T$. Continuing in the same way, the functions $V^2(\overline{\xi}, \xi_T, \tau)$ are determined, and finally

$$V^3(\xi_0, \xi_T, T) = \inf_{\substack{\xi \in \partial D, \\ t \in [0,T]}} \Big\{ c(\xi_0, \xi, t) + V^2(\xi, \xi_T, T - t) \Big\} \tag{3.2}$$

provides the desired value for $M = 3$. Typically, instead of a fixed number M of switches, an upper bound $N \in \mathbb{N}$ on the total number of switches along a hybrid trajectory is given. □

Applying the ideas described in the example above to a general multimodal system, as for example depicted in Figure 3.1, a universally valid but rather vague recurrence relation can be derived:

$$V^K(\xi_1, \xi_2, \tau) = \inf_{\substack{\xi \text{ on a boundary}, \\ t \in [0,\tau]}} \Big\{ c(\xi_1, \xi, t) + V^{K-1}(\xi, \xi_2, \tau - t) \Big\}, \tag{3.3}$$

where the initial condition is given by $V^0(\xi_1, \xi_2, \tau) = c(\xi_1, \xi_2, \tau)$ and $0 < K \leq N$. In the end, the optimal solution associated with the original problem, assuming a given upper bound N on the total number of switches, is obtained by minimizing over all $V^K(\xi_0, \xi_T, T)$, $0 \leq K \leq N$

$$W^N(\xi_0, \xi_T, T) = \min_{0 \leq K \leq N} V^K(\xi_0, \xi_T, T). \tag{3.4}$$

In Chapter 4 and 5, the recursive equation (3.3) is specified more precisely, finally yielding to the so-called Hybrid Bellman Equation.

Subsequently, special emphasis is placed on the noticeable analogy between the backwards Dynamic Programming recursion, derived in Section 2.2.2 to solve the discrete-time optimal control problem, and hybrid Dynamic Programming approach (3.3) for regional dynamics systems. In a careful comparison, the following observations are made:

- The basic concept in both approaches is the definition of an appropriate cost-to-go function which represents the "quality" of a "node" with respect to its

future performance and is used to solve the problem in a backwards manner starting at a given final state and going *back in time* until the initial state is reached. Note that, in the approach presented in this section, the recursion is accomplished by starting with V^0 and continuing the computation with the help of (3.3) until V^N is obtained. Contrarily, in Section 2.2.2, the first cost function is computed for the time stage $k = k_f$ and then k is decreased until $k = k_0$ is reached.

- In the hybrid case, the time-state pairs introduced in Example 3.1 play the role of the states $x(k)$ in the discrete-time problem.

- The fascinating correspondence between the discrete (time) stages k in Section 2.2.2 and the number of switches K along a hybrid trajectory can be observed.

- The cost $c(\cdot, \cdot, \cdot)$ of the non-hybrid state-constrained optimization problem has the same function as the "transition cost" $\ell(\cdot, \cdot, \cdot)$ in the discrete-time problem.

A crucial consequence of the similarity of both approaches is that most of the properties and conclusions mentioned in Section 2.2 and especially in Section 2.2.3 can be adopted directly.

To conclude this section, we return to the starting point of the derivations, i.e. to the division of the complex hybrid point-to-point problem into smaller non-hybrid optimization problems. This procedure establishes a *multistage decision problem* [55], which allows the hybrid optimal control problem to be solved on different levels of abstraction. On a higher level, the problem is approached by using a Dynamic Programming scheme based on the values $c(\cdot, \cdot, \cdot)$ and the cost-to-go functions $V^K(\cdot, \cdot, \cdot)$. However, no attention is paid to the method used to obtain the values $c(\cdot, \cdot, \cdot)$. Moreover, the way of calculating $c(\cdot, \cdot, \cdot)$ does not affect the Dynamic Programming technique in any way. Hence, the computation of $c(\cdot, \cdot, \cdot)$ is part of the lower level of the solution concept. This hierarchic structure is helpful in understanding the complexity of the considered hybrid problem and,

as mentioned, in finally solving the problem. In fact, in Chapter 5 a further level of abstraction, the transition automaton, is introduced.

CHAPTER IV

THE BIMODAL CASE

The purpose of this chapter is to present a first approach to the mathematically precise treatment of the general hybrid point-to-point problem (Section 3.1, 3.2) considering only two regions and limited dynamics which do not allow "bounce back" at a boundary.

Based on the explanatory description of the hybrid optimal control setup (Section 3.1) and the roughly sketched solution scheme (Section 3.2), a Hybrid Bellman Equation for *bimodal* systems is derived revealing the characteristic structure of the hybrid solution and yielding to a computational framework in which numerical results are obtained showing the hybrid behavior under an optimal control policy. The *bimodal* system (with limited dynamics) is considered at first with the intention of illustrating various important mathematical assumptions and definitions needed later to accurately formulate and solve a *general* hybrid point-to-point problem. The bimodal approach represents a straightforward conclusion of Section 3.1 and 3.2 and serves as an introduction to the more complex concepts developed in Chapter 5 in order to approach the general multimodal point-to-point problem, where additionally a precise definition of the discrete control input and the introduction of a further level of abstraction, the transition automaton, is necessary.

The outline of this chapter is as follows: Section 4.1 describes the structure of the partitioned state space including assumptions on the boundaries and on the vector fields associated with each region. The actual hybrid dynamic behavior, especially when reaching a given switching surface, is explained in Section 4.2. Based on these definitions, a mathematically precise formulation of the hybrid point-to-point problem is presented (Section 4.3). The main theorem, the Hybrid

Bellman Equation, is stated in Section 4.4, followed, in Section 4.5, by a discussion on the computational issues concerning the derived recursion equation. Illustrative examples are shown in Section 4.6.

Most of the results stated in this chapter are presented in the 2007 paper, "A Hybrid Bellman Equation for Bimodal Systems", [14].

4.1 The Bimodal System – Regions and Geometric Framework

Important assumptions and definitions regarding the structure of the state space, the dynamics, and, in particular, the relation between the geometry of the regions and the associated dynamic regimes complete the descriptive statements of Section 3.1.

In the bimodal case, *two* open, connected, and simply connected regions D_1 and D_2 with $D_1 \cap D_2 = \emptyset$ are given, forming a partition of the compact state space $\mathcal{X} \subset \mathbb{R}^n$ in the sense that

$$\mathcal{X} = (D_1 \cup \partial D_1) \cup (D_2 \cup \partial D_2), \tag{4.1}$$

where the boundaries $\partial D_i,\ i = 1, 2$ are assumed to be finite unions of closed, smooth codimension one submanifolds s_i^k of $\mathcal{X}$

$$\partial D_i = \bigcup_{k=1}^{n_i} s_i^k, \quad n_i \in \mathbb{N}. \tag{4.2}$$

That is, each boundary ∂D_i is composed of n_i submanifolds s_i^k, $1 \leq k \leq n_i$. The intersection of the boundaries ∂D_1 and ∂D_2 is denoted by ∂D.

With each region $D_i,\ i = 1, 2$ a time-invariant vector field $f_i(x, u)$ is associated which uniquely describes the continuous dynamics in the corresponding partition. However, in order to properly define the hybrid point-to-point problem (Section 4.3), the regional dynamics must satisfy the following assumption, where in the bimodal case $\mathcal{I} = \{1, 2\}$:

Assumption 4.1. *The dynamic behavior within a given contributing region* $D_i,\ i \in \mathcal{I}$ *is defined by*

$$\Sigma_i :\ \dot{x}(t) = f_i(x(t), u(t)), \quad x(t) \in D_i.$$

The continuous-time control input $u(\cdot)$ *of the considered hybrid system lies in the set* $\mathcal{U} = \mathcal{U}\left(U, L_\infty\left([0,T]\right)\right)$ *which contains all bounded measurable functions on the interval* $[0,T]$, $T < \infty$ *taking values in the set* U.

In order to assure existence and uniqueness of the executions in each constituent region D_i, $i \in \mathcal{I}$ *the vector fields* $f_i(x,u)$ *are assumed to be continuously differentiable in* x *(for all* u*) on the closure of* D_i *(and hence uniformly continuous and uniformly Lipschitz in* x *on the closure of* D_i*).*

Furthermore, we assume that the vector field $f_i(x,u)$, $i \in \mathcal{I}$, *satisfies the "transversality" condition in the sense that*

(i) $f_i(x,u)$ *is non-tangential to the boundary* ∂D_i *at any point* $x \in \partial D_i$ *for all choices of* u *and*

(ii) at points $x \in s_i^k \cap s_j^l \neq \emptyset,\ i,j \in \mathcal{I},\ k \in \{1,\ldots,n_i\},\ l \in \{1,\ldots,n_j\}$ *the vector field* $f_i(x,u)$ *is non-tangential to each of the tangent spaces of the intersecting components* s_i^k *and* s_j^l*, for all choices of* u.

The solutions are interpreted in the Carathéodory sense, and the initial condition ξ_0 *of an admissible execution satisfies* $\xi_0 \in D_i$, $i \in \mathcal{I}$. □

An example for the defined bimodal system is depicted in Figure 3.2.

After having specified the geometric framework, the dynamic behavior within the regions, and some important restrictions on the dynamic regimes $f_i(x,u)$ at boundary points, the next step is to describe the transition behavior, that is to say the possible executions and, in particular, to define the dynamics of the bimodal system on the boundaries $\partial D_i,\ i \in \mathcal{I}$.

4.2 The Bimodal System – Dynamics and Executions

Having entered the boundary ∂D from one region, the further execution of the continuous state x, in the considered bimodal system, is *unambiguously* determined by "transitioning through" the boundary and subsequently evolving in the complementary region.

A general explanation of the transition behavior in a regional dynamics system, where both "passing through" and "bouncing back" are possible, is given in Section 3.1 and Figure 3.1 illustrates an example of a possible optimal hybrid trajectory. However, in the presented bimodal approach, the simple case is considered, where the switching behavior is restricted to the "transitioning through" option; hence, only case (i) in Section 3.1 can occur. Consequently, the further evolution of the continuous state x arrived at the boundary ∂D is clearly defined – without any ambiguity:

In order to precisely describe the transition behavior, the notation x_k, where $\dot{x}_k(t) = f_k(x_k(t), u(t)),\ k \in \mathcal{I} = \{1, 2\}$, is used to emphasize which dynamic regime f_k determines the execution of the trajectory at a given point $x_k(t) \in D_k \cap \partial D_k$. Assuming a given starting state $\xi_0 \in D_i,\ i \in \mathcal{I}$, of the continuous state's evolution, the control input $u(\cdot) \in \mathcal{U}$ gives rise to a trajectory evolving according to $\dot{x}_i = f_i(x_i, u)$. If there is a finite time $t_s < T$ such that the state x enters the boundary ∂D, that is

$$\xi_s = \lim_{t \to t_s} x_i(t) \ \in \partial D,$$

then, at that time, a switch "through" the boundary occurs and the continuous state x evolves afterwards in the complementary region j with the dynamics $\dot{x}_j = f_j(x_j, u)$ and the initial condition $x_j(t_s) = \xi_s$, where $j = 1$ if $i = 2$ and $j = 2$ if $i = 1$, until a further possible intersection with the boundary ∂D. Therefore, as depicted in Figure 3.2, the location of the continuous state's evolution, displayed by the solid blue line, alternates between region 1 and 2.

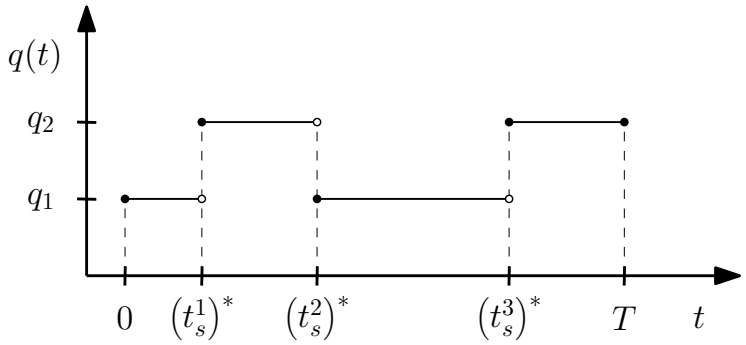


Figure 4.1: Discrete-State Trajectory Corresponding To The Continuous State's Evolution In Figure 3.2

With the intention of specifying the actual region of the continuous state $x(t)$, a discrete state $q(t)$ is introduced taking values from the discrete state space $\mathcal{Q} = \{q_1, q_2\}$, which is in one-to-one correspondence to the set of regions $\mathcal{D} = \{D_i : \ i \in \mathcal{I}\}$. Obviously, in case the continuous (valued) state $x(t)$ lies in the interior of D_i the corresponding discrete state is $q(t) = q_i$. However, when $x(t)$ lies on the boundary ∂D the interpretation of the possible discrete state values $q(t-) = q_1$ or $q(t-) = q_2$ is that the continuous state has arrived at $x(t)$ along a trajectory which most recently lay in D_1, or respectively, D_2, and a switch of the discrete state to $q(t) = q_j$, where $j = 2$, or respectively, $j = 1$, indicates that the system trajectory "passes through" the boundary and will evolve in the complementary region D_j under the jth vector field on a time interval with initial instant t.

In brief, the description of the hybrid execution includes both continuous- and discrete-valued states, where the latter have right-continuous trajectories in $\mathbb{R}$ which are piecewise constant. Figure 4.1 shows the discrete-valued state $q(t), \ t \in [0, T]$ corresponding to the continuous state's evolution depicted in Figure 3.2. Additionally, from the previous specifications it follows that over the finite interval $[0, T]$, with $x(0) = \xi_0 \in D_i$ for some $i \in \mathcal{I}$, the resulting controlled trajectories $x(\cdot)$ are continuous for any control function $u(\cdot) \in \mathcal{U}$.

Especially noteworthy is also that, as a consequence of the restricted bimodal dynamics, the behavior on the boundary is clearly determined, that is, no decision

is to be made if the continuous state x arrives at the boundary. As can be seen in Chapter 5, the solution of the general hybrid point-to-point problem, where both phenomena, "passing through" and "bouncing back", can occur, involves the introduction of a (controlled) discrete input specifying the action at a certain switching point. However, in order to make this first approach more accessible the discrete input events are avoided by the presented precise assumptions on the bimodal transition behavior.

Based on the previous definitions, the following section formally introduces the hybrid point-to-point optimization problem.

4.3 The Optimization Problem

The explanatory problem formulation in Section 3.1, combined with the assumptions given in Section 4.1 and the bimodal dynamics presented in Section 4.2, yields to an exact formulation of the considered hybrid point-to-point problem.

To begin with, the complementary indicator $q^c(t) \in \mathcal{Q} = \{q_1, q_2\}$ is introduced, which gives in each case, $q(t) = q_1$ and $q(t) = q_2$, the opposite (discrete) state, $q^c(t) = q_2$ and $q^c(t) = q_1$, respectively. With this definition and a given *upper bound $N \in \mathbb{N} \cup \{0\}$ on the total number of switches along the optimal trajectory*, the hybrid optimization problem, addressed in this paper and previously illustrated in Section 3.1, is stated as follows:

Problem 4.1 (The Bimodal Point-To-Point Problem).

$$\mathcal{P}_N : \inf_{u(\cdot)\in\mathcal{U}} \int_0^T \ell(x(t), u(t))\, dt \tag{4.3}$$

subject to

- *the discrete-state dynamics*

$$q(t) = q^c(t_s^k-), \quad t \in \left[t_s^k, t_s^{k+1}\right), \tag{4.4}$$

if $1 \leq k \leq M$ and the initial region is given by $q(t) = q_{i_0}$, $t \in [0, t_s^1)$,

- *the continuous-state dynamics*

$$\dot{x}_{i_k}(t) = f_{i_k}(x_{i_k}(t), u(t)), \quad t \in \left[t_s^{k-1},\, t_s^k\right), \tag{4.5}$$

where $0 < k \leq M+1$, $x_{i_k}(t) \in \mathcal{X}$, *and* i_k *is uniquely specified by the discrete dynamics* (4.4),

- *and the corresponding initial and final conditions*

$$\begin{aligned} x(0) &= x_{i_1}(t_s^0) &&= \xi_0 \quad \in D_{i_0}, \\ x_{i_{k+1}}(t_s^k) &= \lim_{t \to t_s^k} x_{i_k}(t) &&= \xi_s^k \quad \in \partial D, \\ x(T) &= x_{i_{M+1}}(t_s^{M+1}) &&= \xi_T \quad \in D_{i_T}, \end{aligned} \tag{4.6}$$

if $0 < k \leq M$ *and* $i_0, i_T \in \mathcal{I}$.

The number of switches M *along the trajectory satisfies* $0 \leq M \leq N$ *and is even if* $i_0 = i_T$ *and odd if* $i_0 \neq i_T$. □

Note that $\xi_0, \xi_T \notin \partial D$ and $(t_s^k,\, \xi_s^k)$, $0 < k \leq M$ denotes the kth switching point along the trajectory.

The subsequent section derives, based on the above problem formulation, a recursion relation solving the bimodal point-to-point problem.

4.4 The Hybrid Bellman Equation

Along with the illustrations in Section 3.2, a Hybrid Bellman Equation for bimodal systems is obtained providing a characterization of global optimality.

Recalling the definition of the discrete variable $q(t) \in \mathcal{Q}$, which assigns a region, and concurrently also a governing dynamical system, to each continuous state x along a hybrid trajectory, the functions $c(\cdot, \cdot, \cdot)$ and $V^{\cdot}(\cdot, \cdot, \cdot)$ introduced in Section 3.2 are defined more precisely:

Definition 4.1. *The infimum of the costs associated with driving the system from* $\xi_1 \in D_{i_1} \cup \partial D_{i_1}$ *to* $\xi_2 \in D_{i_1} \cup \partial D_{i_1}$, $i_1 \in \mathcal{I}$ *over a time horizon* Δ *without leaving* $D_{i_1} \cup \partial D_{i_1}$ *and without a switching taking place is denoted* $c(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \Delta)$,

where $q_{i_1} \in Q$ and $q_{i_2} \in Q$ represent the discrete states associated with points ξ_1 and ξ_2, respectively. Clearly, the cost $c(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \Delta)$ is infinite if ξ_2 is inaccessible from ξ_1 by trajectories remaining in $D_{i_1} \cup \partial D_{i_1}$ and along which no switch occurs. □

Definition 4.2. *The cost-to-go function $V^M(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau)$ is defined as the infimum of the costs of going from $\xi_1 \in \mathcal{X}$ to $\xi_2 \in \mathcal{X}$ during the time horizon τ using exactly M switches and starting in region D_{i_1}. Again, the values $q_{i_1} \in Q$ and $q_{i_2} \in Q$ represent the discrete states associated with the points ξ_1 and ξ_2, respectively. If ξ_2 cannot be reached from ξ_1 under the given specifications, $V^M(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau)$ is infinity.* □

By the preliminary sections, Section 2.2 and 3.2, a profound background and a detailed explanation of our general idea to solve the hybrid point-to-point problem is given, which allows now to establish the main theorem as a direct consequence. On the basis of the previously defined functions $c(\cdot, \cdot, \cdot, \cdot, \cdot)$ and $V^{\cdot}(\cdot, \cdot, \cdot, \cdot, \cdot)$, the Hybrid Bellman Equation for bimodal systems is stated as follows:

Theorem 4.1 (The Hybrid Bellman Equation for Bimodal Systems). *Assume that all hypotheses for the existence and uniqueness of the bimodal executions hold and that all infima exist in the definition of the hybrid value functions $V^{\cdot}(\cdot, \cdot, \cdot, \cdot, \cdot)$, for all admissible argument values, whenever the expressions are finite. Then, the recurrence relation is expressed by*

$$V^K(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau) = \inf_{\substack{\xi \in \partial D, \\ t \in [0,\tau]}} \Big\{ c(\xi_1, q_{i_1}, \xi, q^c_{i_1}, t) + V^{K-1}(\xi, q^c_{i_1}, \xi_2, q_{i_2}, \tau - t) \Big\}. \quad (4.7)$$

This relation holds as long as $K \geq 1$. If $K = 0$, the following direct simplification is obtained

$$V^0(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau) = c(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau). \quad (4.8)$$

□

Since the bimodal point-to-point problem (Problem 4.1) does not insist on an *a priori* given number of switches, it is finally necessary to relate $V^K(\xi_1,\, q_{i_1},\, \xi_2,\, q_{i_2},\, \tau)$ to the original problem, cf. Equation (3.4). With the boundary conditions, $x(0) = \xi_0 \in D_{i_0}$ and $x(T) = \xi_T \in D_{i_T},\ i_0, i_T \in \mathcal{I}$, and the given upper bound N on the total number of switches along a hybrid trajectory, the optimal cost $W^N(\xi_0,\, q_{i_0},\, \xi_T,\, q_{i_T},\, T)$ corresponding to the initially defined problem is given by

$$W^N(\xi_0,\, q_{i_0},\, \xi_T,\, q_{i_T},\, T) = \min_{0\leq K\leq N} V^K(\xi_0,\, q_{i_0},\, \xi_T,\, q_{i_T},\, T). \tag{4.9}$$

Illustrated in the subsequent section is the actual computational procedure involved in solving the bimodal point-to-point problem based on the derived recurrence relation (4.7).

4.5 Implementation and Computational Issues

In brief, two main characteristics are associated with the recursive algorithm proposed in Theorem 4.1: first, a large preliminary computational effort – allowing, however, fast solutions afterwards – and, second, the discretization of the boundary distinguishing our method from other approaches which actually insist on discretizing the entire state space $\mathcal{X}$, see e.g. [6, 28].

In continuing this section, the procedure of computationally performing the recursion scheme, given in Theorem 4.1, is illustrated, where special emphasis is placed on the discretization needed to numerically solve the hybrid point-to-point problem. Moreover, of peculiar interest is also the efficiency of the proposed (Dynamic Programming) algorithm. At the end, it remains to say a few words about the particular method used in Section 4.6 to exemplarily solve some interesting bimodal point-to-point problems.

A helpful starting point for the following considerations is the explanatory description in Section 3.2 and, particularly, in Example 3.1, which explicitly points out the necessity of calculating the functions $c(\xi_1,\, q_{i_1},\, \xi_2,\, q_{i_2},\, \tau)$ between all pairs $(\xi_1, \xi_2) \in \partial D \times \partial D$ for all $\tau \in [0, T]$ and $i_1 \in \mathcal{I} = \{1, 2\}$ in order to accomplish the recursion routine (4.7). Actually, these functions can be computed prior to the

backwards minimization procedure, since they depend neither on the intermediate results of (4.7) nor on the boundary conditions ξ_0 and ξ_T. This fact highlights again the hierarchical structure of the hybrid point-to-point problem consisting of a lower level of abstraction which includes the calculation of the functions $c(\cdot, \cdot, \cdot, \cdot, \cdot)$ between any two boundary points and a higher level represented by the Hybrid Bellman Equation (4.7). As regards the hierarchical decomposition of the problem, supplementary remarks are made in Section 2.2 and Section 3.2, where the hybrid optimal control problem is characterized as a *multistage decision problem* [55] and further interesting properties concerning, in particular, the solution procedure on different levels of abstraction are mentioned.

From the preceding introductory comments and with a careful consideration of the recurrence relation given in Theorem 4.1, the computational routine solving the bimodal point-to-point problem follows immediately:

The first step in the course of computation is the calculation of the costs $c(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau)$ for all pairs $(\xi_1, \xi_2) \in \partial D \times \partial D$ with $\tau \in [0, T]$ and $i_1 \in \mathcal{I} = \{1, 2\}$. Note that the cost $c(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau)$ is not affected by the discrete state q_{i_2} associated with the final point ξ_2, cf. Definition 4.1. Indeed, the actual calculation of the costs $c(\cdot, \cdot, \cdot, \cdot, \cdot)$ is performed by discretizing the boundary ∂D as well as the time interval $[0, T]$ and solving for all discrete combinations

$$(\xi_1^d, \xi_2^d, \tau^d, i_1), \qquad i_1 \in \mathcal{I}, \tag{4.10}$$

where τ^d lies in the finite set of time values $\mathcal{T}_{discrete} \subset [0, T]$ and ξ_1^d, ξ_2^d are elements from the set $\mathcal{D}_{discrete} \subset \partial D$ consisting of a finite selection of boundary points. The notation $(\cdot)^d$ points out that the variable $(\cdot)$ takes values from a finite set of elements. For the purpose of estimating the computational effort associated with these preliminary calculations, the boundary ∂D is supposed to be given by

$$\partial D = \Big\{ x \in \mathbb{R}^n \;\Big|\; x_i \in [\,\alpha_i, \beta_i\,],\ i \in \{1, 2, \ldots, n-1\}\,; \\ x_n = g(x_1, x_2, \ldots, x_{n-1}) \Big\}. \tag{4.11}$$

With the definition (4.11) and under the assumption that the intervals $[\,\alpha_i, \beta_i\,]$, $i \in \{1,\, 2,\, \ldots,\, n-1\}$ are discretized into $(N_x - 1)$ subintervals, leading to a finite set of discrete values $\left\{\alpha_i,\, x_i^1,\, x_i^2,\, \ldots,\, x_i^{N_x-2},\, \beta_i\right\}$ for each direction i, the discrete combinations $(\xi_1^d,\, \xi_2^d,\, \tau^d,\, i_1) \in \mathcal{D}_{discrete}^2 \times \mathcal{T}_{discrete} \times \mathcal{I}$ are uniquely specified by the grid $\mathcal{G}$

$$\mathcal{G} = \left(\times_{i=1}^{n-1} \left\{\alpha_i,\, x_i^1,\, x_i^2,\, \ldots,\, x_i^{N_x-2},\, \beta_i\right\}\right)^2 \times \mathcal{T}_{discrete} \times \{q_1,\, q_2\}\,, \tag{4.12}$$

where $|\mathcal{T}_{discrete}| = N_\tau$ is assumed. As a consequence, the number of costs $c(\cdot,\, \cdot,\, \cdot,\, \cdot,\, \cdot)$, to be calculated initially, is given by

$$n_{cost} = 2N_\tau \left(N_x^{n-1}\right)^2 \tag{4.13}$$

and, hence, the effort associated with this first computational step increases *linearly* when raising N_τ, *polynomially* when raising N_x, and *exponentially* when raising the dimensionality n of the state space $\mathcal{X}$. The latter results in a time-consuming computation for high-dimensional systems as, for example, observed in Chapter 7.

In fact, as illustrated e.g. in Figure 3.2, except for the initial and final pieces of the hybrid optimal trajectory, that is, from ξ_0 to the first intersection of the switching boundary and from the last intersection to ξ_T, the optimal hybrid execution is simply given by a concatenation of trajectories between points on the boundary. Consequently, the only computation, which still needs to be done before starting the recursion algorithm (4.7), is the evaluation of the costs $c(\xi_0,\, q_{i_0},\, \xi_2^d,\, q_{i_2},\, \tau^d)$ and $c(\xi_1^d,\, q_{i_1},\, \xi_T,\, q_{i_T},\, \tau^d)$, where $\tau^d \in \mathcal{T}_{discrete}$, $\xi_1^d, \xi_2^d \in \mathcal{D}_{discrete}$, and $i_1 \in \mathcal{I}$. This step obviously depend on the given initial and final states, ξ_0 and ξ_T, whereas the calculations before are independent of the boundary conditions.

After the costly preparation of all relevant values $c(\cdot,\, \cdot,\, \cdot,\, \cdot,\, \cdot)$, the minimizations in (4.7), associated with the actual Dynamic Programming recursion of Theorem 4.1, can be easily accomplished. As for the computational effort inherent in the recursive algorithm, the comments made in Section 2.2.3 on the

(discrete-time) Bellman equation apply also for the recently derived *Hybrid* Bellman Equation (4.7). The analogy between both approaches, previously illustrated in Section 3.2, immediately allows some interesting conclusions on the numerical properties of the proposed hybrid algorithm (Theorem 4.1). Roughly speaking, in the hybrid backwards recursion procedure, the pairs $(\xi_1^d,\, \tau^d) \in \mathcal{D}_{discrete} \times \mathcal{T}_{discrete}$ play the same role as the discrete states $x(k)$ in the original Bellman equation (2.15). Taking that analogy a step further, the number of switches K along a hybrid trajectory, cf. equation (4.7), corresponds to the discrete (time) stages k in (2.15). However, considering the Hybrid Bellman Equation (4.7), the minimization is formulated over the time-state pairs $(\xi,\, t) \in \partial D \times [0, \tau]$ instead of the input segments $\{u(s) \in \mathbb{R}^m \mid\ s \in [0, t\,]\,\}$, where $t \in [0, \tau]$. Consequently, comparing the original Bellman Equation (2.15) with the discretized version of (4.7), the set $\mathcal{U}_{ad}(m, x(m))$ is "replaced by" a set

$$\left\{ \left(\xi^d, t^d\right) \;\mid\; \left(\xi^d, t^d\right) \in \mathcal{D}_{discrete} \times \mathcal{T}_{discrete}\,,\ 0 \leq t^d \leq \tau \right\}. \tag{4.14}$$

Based on the previously stated analogies, a rough estimate on the computational effort associated with the bimodal Dynamic Programming approach, Theorem 4.1, is obtained by approximately estimating the numbers a, b, and c, defined in Section 2.2.3,

$$a \approx N_\tau N_x^{n-1}, \quad b \approx N_\tau N_x^{n-1}, \quad \text{and} \quad c = K. \tag{4.15}$$

Recalling the relations (2.18) and (2.19) presented in Section 2.2.3, the following conclusions can be made on the computational effort associated with the bimodal recursion algorithm, cf. (4.15): the computational effort increases *quadratically* when raising N_τ and *polynomially* when raising N_x. Moreover, the mentioned "curse of dimensionality" [4], Section 2.2.3, describing the *exponential* growth of the computational effort when increasing the dimensionality n of the state space $\mathcal{X}$, is also inherent in the hybrid approach as can be seen by using (4.15) in equations (2.18), (2.19).

In summary, a computational framework is established, in which a large computational burden is needed initially when preparing the costs $c(\cdot,\, \cdot,\, \cdot,\, \cdot,\, \cdot)$, but once that price is paid, fast solutions are possible using the Hybrid Bellman Equation (4.7). In fact, the calculation of the costs $c(\xi_1,\, q_{i_1},\, \xi_2,\, q_{i_2},\, \tau)$ between pairs of boundary points $(\xi_1, \xi_2) \in \partial D \times \partial D$ can be interpreted in terms of optimality zones. Refer to [15] for a detailed introduction to this topic. In addition, especially noteworthy is that the introduced discretization, necessary to numerically solve the bimodal point-to-point problem, might have a huge influence on the actual optimal solution obtained by the previously described computational procedure. Indeed, only the discrete state-time pairs $(\xi^d,\, t^d) \in \mathcal{D}_{discrete} \times \mathcal{T}_{discrete}$ are possible choices for switching points (ξ_s^k, t_s^k) along a hybrid trajectory when performing the recursion algorithm (4.7). Furthermore, it is implicitly assumed that the envelope of $\mathcal{D}_{discrete}$ contains the optimal switching points occurring along a hybrid optimal execution. As a consequence, the discretization of both the boundary ∂D and the time interval $[0, T]$ must be chosen very carefully.

The last comments of this section are made on the particular computational methods used in Example 4.1 and 4.2, Section 4.6, to calculate the hybrid optimal trajectories of two illustrative bimodal point-to-point problems. As emphasized previously, the minimizations (4.7) on the higher level of abstraction depend entirely on the subordinate calculation of the costs $c(\cdot,\, \cdot,\, \cdot,\, \cdot,\, \cdot)$, which represents a non-hybrid state-constrained optimization problem. In the subsequent examples, however, only bimodal systems or, more specifically, only bimodal point-to-point problems (Problem 4.1) are considered for which the *state constraints are inactive along the hybrid optimal trajectory.* More precisely, the following assumption is satisfied:

Assumption 4.2. *Along a hybrid optimal trajectory $x^*(\cdot)$ associated with a particular bimodal point-to-point problem, Problem 4.1, the individual contributing segments between two consecutive points $\left(t^k, x^k\right)$, $\left(t^{k+1}, x^{k+1}\right)$, $0 \leq k \leq M^*$ of*

the sequence

$$\begin{aligned}&\big((0,\xi_0),\,(t_s^1,\xi_s^1)^*,\,(t_s^2,\xi_s^2)^*,\,\ldots,\,(t_s^{M^*},\xi_s^{M^*})^*,\,(T,\xi_T)\big)\\&\qquad=\big((t^0,x^0),\,(t^1,x^1),\,(t^2,x^2),\,\ldots,\,(t^{M^*},x^{M^*}),\,(t^{M^*+1},x^{M^*+1})\big)\,,\end{aligned}\tag{4.16}$$

where $(t_s^m,\xi_s^m)^*\in(0,T)\times\partial D,\ i\in\{1,2,\ldots,M^*\}$ *represents the time-state pair of the* m*th switch along the optimal hybrid trajectory and* M^* *denotes the optimal number of switches, coincide with the trajectories obtained by solving the corresponding unconstrained optimal control problem between the respective states* $\big(t^k,x^k\big)$ *and* $\big(t^{k+1},x^{k+1}\big)$*:*

$$\mathcal{P}:\inf_{u\in\mathcal{U}}\int_{t^k}^{t^{k+1}}\ell(x(t),u(t))\,dt\tag{4.17}$$

subject to $\dot{x}(t)=f_i\left(x\left(t\right),u\left(t\right)\right)$ *with the initial condition* $x(t^k)=x^k$ *and the final condition* $x(t^{k+1})=x^{k+1}$*, where* $i=1$ *if the original segment of the hybrid optimal trajectory lies in region* D_1 *and* $i=2$ *otherwise.* □

As a consequence of Assumption 4.2, the functions $c(\cdot,\,\cdot,\,\cdot,\,\cdot,\,\cdot)$ preparing the Dynamic Programming algorithm can be considered in an ordinary *unconstrained* optimal control framework. In fact, in the case of linear dynamics and a quadratic cost function as chosen in Example 4.1 and 4.2, the approach presented in Section 2.3.2 or, more specifically, equation (2.40) yield to the desired costs $c(\cdot,\,\cdot,\,\cdot,\,\cdot,\,\cdot)$. Since the key objective of the following numerical computations is to investigate the efficiency and computability of the recursive scheme associated with the derived Hybrid Bellman Equation (4.7), Assumption 4.2 represents a reasonable simplification avoiding the implementation of a routine for solving the actual state-constrained problem $c(\cdot,\,\cdot,\,\cdot,\,\cdot,\,\cdot)$. However, investigations dealing with state-constrained optimization problems can be found in [7, 17, 21, 22, 26, 30, 53].

4.6 Examples

Two examples are presented with the objective of both highlighting the operation of the proposed Dynamic Programming approach and illustrating, concurrently,

the dynamic behavior of the introduced bimodal systems subjected to an optimal control policy. The examples are chosen in such a way that Assumption 4.2 holds. Consequently, the relevant costs $c(\cdot, \cdot, \cdot, \cdot, \cdot)$ are simply given by (2.40).

First, a bimodal framework is considered showing exactly the same optimal control parameters as the non-hybrid point-to-point problem in Example 2.2. However, in the second half of the state space, a different dynamical regime determines the continuous states' execution. Hence, this setup is ideally suited to explore the hybrid character of the bimodal solution compared to the ordinary one-regional solution depicted in Figure 2.4(b).

Example 4.1. In this example, the planar state space $\mathcal{X} \subset \mathbb{R}^2$ is divided into the regions $D_1 = \{x \in \mathbb{R}^2 \mid (1 \;\; 1)\, x > 0\}$ and $D_2 = \{x \in \mathbb{R}^2 \mid (1 \;\; 1)\, x < 0\}$. The regional dynamics determining the hybrid behavior of the continuous state $x(t) = \begin{pmatrix} x_1(t) & x_2(t) \end{pmatrix}' \in \mathcal{X}$ are given by

$$\dot{x}(t) = \begin{cases} \begin{pmatrix} 0.8 & 1 \\ -3 & -5 \end{pmatrix} x(t) + \begin{pmatrix} -0.3 \\ 3 \end{pmatrix} u(t), & x(t) \in D_1 \\ \begin{pmatrix} -0.3 & 0.05 \\ -0.5 & 0 \end{pmatrix} x(t) + \begin{pmatrix} 0.1 \\ 1 \end{pmatrix} u(t), & x(t) \in D_2, \end{cases} \tag{4.18}$$

where the dynamical regime in region D_1 corresponds to the governing dynamics (2.45) in Example 2.2. Furthermore, the boundary conditions, $x(0) = \xi_0 = (-1 \;\; 0)'$ and $x(3) = \xi_T = (1 \;\; 0)'$, as well as the cost function (4.3), specified by $\ell(x(t), u(t)) = u(t)^2$, are identical in both examples. With an unconstrained input $\mathcal{U} = \mathbb{R}$ and the maximum number of intersections being given by $N = 20$, the bimodal point-to-point problem, Problem 4.1, is fully defined.

In fact, the numerical solution is obtained by discretizing the time interval $\mathcal{T} = [0, 3]$ into 60 equally spaced temporal steps and the boundary $\partial D : (1 \; 1)\, x = 0$ into 40 equally spaced spatial steps over the interval $x_1 \in [-3, 3]$. Figure 4.2(a) shows the resulting optimal trajectory $(x_1^*(\cdot), x_2^*(\cdot))$ in blue (solid line); the dashed line represents the boundary ∂D and the circles depict the particular switching

points. In this case, the optimal solution is obtained, when only one crossing of the boundary takes place, with the corresponding optimal cost (4.9) being

$$W^{20}(\xi_0,\, q_2,\, \xi_T,\, q_1,\, 3) = V^1(\xi_0,\, q_2,\, \xi_T,\, q_1,\, 3) \approx 14.99. \tag{4.19}$$

Finally, it is interesting to compare the optimal trajectory corresponding to the point-to-point problem solved in one region, Figure 2.4(b), with the optimal execution under a bimodal regime (4.18) as depicted in Figure 4.2(a). □

In order to highlight the fact that multiple switches may be to prefer, another linear situation is considered:

Example 4.2. A bimodal system under the regional dynamics

$$\dot{x}(t) = \begin{cases} \begin{pmatrix} \varepsilon_{11} & \omega_1 \\ -\omega_1 & \varepsilon_{12} \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(t), & x(t) \in D_1 \\ \begin{pmatrix} \varepsilon_{21} & \omega_2 \\ -\omega_2 & \varepsilon_{22} \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(t), & x(t) \in D_2, \end{cases} \tag{4.20}$$

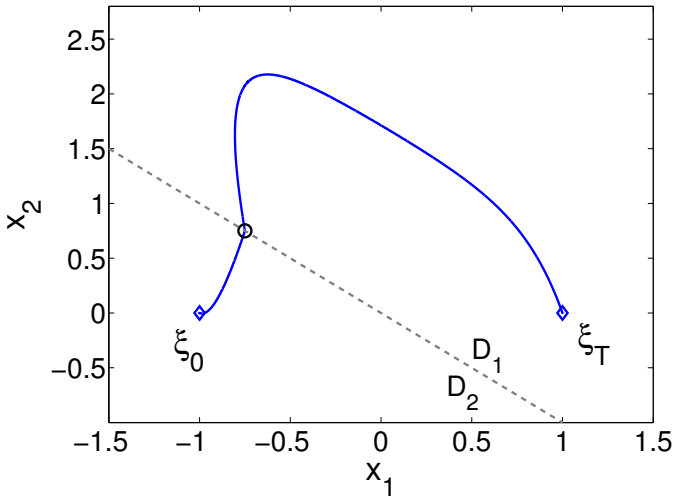
with $D_1 = \{x \in \mathbb{R}^2 \mid (1 \;\; 1)\, x > 0.5\}$ and $D_2 = \{x \in \mathbb{R}^2 \mid (1 \;\; 1)\, x < 0.5\}$, is driven between the points $x(0) = \xi_0 = (-0.5 \;\; 0)'$ and $x(T) = \xi_T = \left(1/\sqrt{8} \;\; 1/\sqrt{8}\right)'$, where

$$\omega_1 = \frac{\pi}{2}, \qquad \omega_2 = \frac{\pi}{4}, \quad \text{and} \quad T = \frac{\pi}{2\omega_2} + \frac{\pi}{2\omega_1} + \frac{3\pi}{2\omega_2} + \frac{\pi}{4\omega_1} = 9.5. \tag{4.21}$$

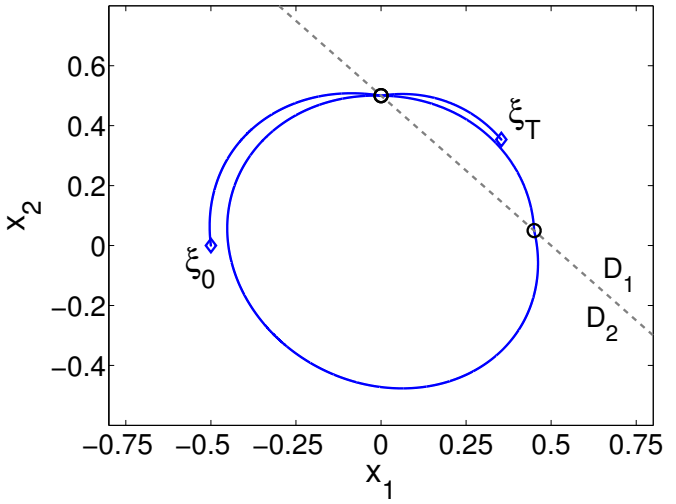
As in the previous example, the cost function under consideration is the control energy of the control signal, that is, $\ell(x(t), u(t)) = u(t)^2$ and, again, the input $u(t)$ is supposed to be unconstrained, i.e. $u(t) \in \mathcal{U} = \mathbb{R},\ t \in \mathcal{T} = [0, T]$.

Actually, if $\varepsilon_{ij} = 0,\ i, j \in \{1, 2\}$, a zero control effort would result in a three-switch situation. By making $\varepsilon_{11} = \varepsilon_{12} = -0.1$ and $\varepsilon_{21} = \varepsilon_{22} = -\varepsilon_{11}$, a slightly disturbed oscillation dynamic is obtained in each region. Using exactly the same discretization parameters as in the example above, the three-switch situation is still optimal, cf. Figure 4.2(b), with

$$W^{20}(\xi_0,\, q_2,\, \xi_T,\, q_1,\, 3) = V^3(\xi_0,\, q_2,\, \xi_T,\, q_1,\, 3) \approx 0.043. \tag{4.22}$$

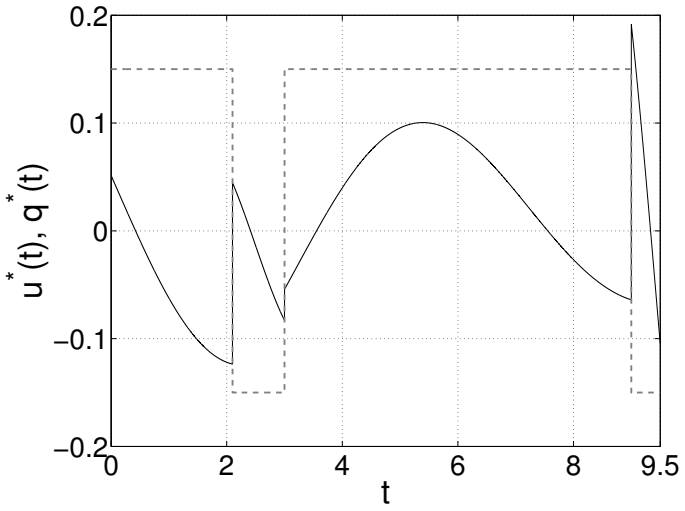


(a) One Switch is Optimal.

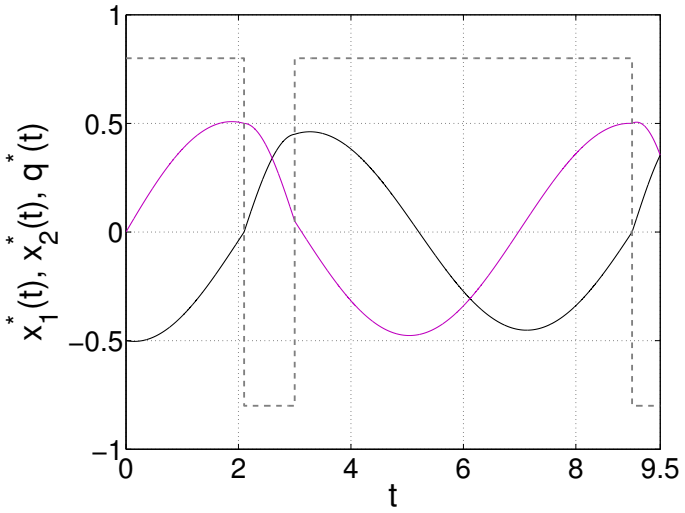


(b) Three Switches are Optimal.

Figure 4.2: Optimal Trajectories $(x_1^*(\cdot), x_2^*(\cdot))$ Corresponding To Example 4.1 And Example 4.2, Respectively.



(a) Optimal Input $u^*(\cdot)$



(b) Optimal States $x_1^*(\cdot)$ and $x_2^*(\cdot)$, black line and purple line, respectively

Figure 4.3: Optimal Trajectories Corresponding To Example 4.2

Figure 4.3 depicts both the optimal input $u^*(\cdot)$ and the optimal states $x_1^*(\cdot)$, $x_2^*(\cdot)$ over the time interval $\mathcal{T} = [0, 9.5]$. Here, the dashed gray line represents the discrete state $q^*(\cdot)$, where the upper level corresponds to the value q_2 and the lower level to q_1. □

CHAPTER V

THE MULTIMODAL SYSTEM

This chapter completes the previous considerations of Section 3.1, Section 3.2, and Chapter 4 by finally presenting global optimality conditions for the general multimodal point-to-point problem, which is characterized by an arbitrary large, but finite, number of regions and the possibility of "passing through" or "bouncing back" at a boundary.

The subsequent approach represents a mathematically precise treatment of the informally introduced multiregional point-to-point problem (Section 3.1) and its briefly adumbrated solution scheme (Section 3.2). However, considering the outline of this work, the results in this chapter can also be regarded as a straight continuation of the work done in Chapter 4 approaching the bimodal point-to-point problem. The following derivations are primarily driven by these preliminary discussions and provide a notable extension and generalization of the conclusions in Chapter 4. Based on the definitions and assumptions introduced in the bimodal case, a similar strategy is pursued finally yielding to a Hybrid Bellman Equation for multimodal systems. Compared to the bimodal solution scheme, Theorem 4.1, the Dynamic Programming algorithm solving the general multiregional point-to-point problem is particularly characterized by the following two features neglected in the precedent simplified bimodal procedure, which considered only two regions and limited dynamics.

(i) One novelty in the subsequent considerations is the advanced geometric structure, namely, a state space $\mathcal{X}$ which is divided into multiple, arbitrary connected regions. In order to solve the hybrid point-to-point problem in this structural framework, an important step is to establish an appropriate representation of the composed state space $\mathcal{X}$. This goal is accomplished

by introducing an additional level of abstraction, the so-called transition automaton, which contains the entire global knowledge of the given geometry and, therefore, provides information on the connections between the individual regions and on possible sequences of transitions leading from one region to another. At this point, an important remark should be made on the previously considered bimodal point-to-point problem in Chapter 4, which considers only one specific regional arrangement. As a consequence, in this case, the global information on the geometric structure is directly included in the Hybrid Bellman Equation, cf. (4.7), and the additional level of abstraction can be avoided.

(ii) Furthermore, the dynamic restrictions, assumed in the bimodal approach (Section 4.2), are lifted, which means that both transition characteristics illustrated in Section 3.1, "passing through" and "bouncing back", are admissible at boundary points, forcing the definition of a discrete control input, which uniquely specifies the behavior at a certain switching point. Consequently, the optimal solution of the multimodal point-to-point problem is hyrbid in nature in that it depends not only on the continuous control signal $u(\cdot)$, but also on a discrete (control) input sequence specifying the domains the system should go through in the first place.

The outline of this chapter reflects the analogy between the bimodal approach (Chapter 4) and the way of proceeding in the general multimodal case. Starting, in Section 5.1, with an illustration of the hierarchic structure inherent in the considered multimodal point-to-point problem, the following sequence of sections runs completely parallel to the endeavor in Chapter 4: Based on the preliminary results for bimodal systems, in a first step, a formal description of the regional dynamics system is introduced. This is done by a discussion on the geometric structure of the partitioned state space (Section 5.2) and a precise formulation of the hybrid dynamic behavior (Section 5.3). The following section, Section 5.4, is devoted to the new level of abstraction, the transition automaton, encoding

the way in which the system can transition from one region to another. Against this background, Section 5.5 presents the mathematically precise formulation of the hybrid point-to-point problem and the main theorem, the Hybrid Bellman Equation for multimodal systems, is stated in Section 5.6. This is followed, in Section 5.7, by some comments on the computational effort needed to accomplish the derived recursion algorithm. Finally, in Section 5.8, a number of examples are presented to highlight the operation of the proposed approach.

This chapter represents a continuation and extension of the developments begun in [14] and the results of the following sections are summarized in the paper, "A Hybrid Bellman Equation for Systems with Regional Dynamics", [58].

5.1 The Hierarchic Structure of The Problem

This section provides a big-picture look at the general point-to-point problem delivering insight into the problem's structural composition, which represents the underlying framework of the proposed solution scheme illustrated later by Section 5.4 and Section 5.6.

As already indicated in the introductory sections, Section 2.2.3 and Section 3.2, a clear analysis and a careful structuring of the hybrid point-to-point problem offer a deep understanding of the problem's complexity and, even more important, also an advantageous starting point for deriving a solution concept yielding to a characterization of global optimality. Here, the novelty lies in the interpretation of the hybrid point-to-point problem as a *multistage decision problem* [55], which creates a hierarchical structure consisting of different levels of abstraction and makes it possible to apply an adequate Dynamic Programming algorithm, cf. Section 3.2 and Section 4.4.

In this work, the general multimodal point-to-point problem is approached on three different levels of control: On the highest level, the given geometric framework is taken into account. The introduced transition automaton, a discrete representation of the composed state space $\mathcal{X}$, specifies the connections between

the different regions and the associated language provides all sequences of transitions, which are possible in the given partitioned state space $\mathcal{X}$. Furthermore, the subsequent section, Section 5.4, illustrates that, on this level of control, arbitrary switching rules can be incorporated by arranging the connections between the automaton's discrete states in an appropriate way.

At a level below, based on the language generated by the automaton a Hybrid Bellman Equation is derived. This equation provides, for an *a priori* given number M of switches along the hybrid trajectory, the *optimal* switching points $(t_s^i, \xi_s^i)^*, \quad i \in \{1, 2, \ldots, M\}$ together with the corresponding sequence of domains, the continuous state x goes through. In addition, the associated cost $V^M(\xi_0, q_{i_0}, \xi_T, q_{i_T}, T)$, as defined in Definition 4.2, is provided. However, in order to solve the original problem, where an upper bound N on the total number of switches is given (cf. Section 3.2 and Section 4.3), a minimization over all $V^M(\xi_0, q_{i_0}, \xi_T, q_{i_T}, T)$, $0 \leq M \leq N$, cf. equation (4.9), is necessary leading to an optimal number of switches M^* associated with an optimal sequence of switching points and the corresponding optimal series of passed regions.

Using this result, the actual optimal path connecting the calculated switching points and the initial and final point, $\xi_0 \in D_{i_0}$ and $\xi_T \in D_{i_T}$, respectively, is obtained by separately solving – on the lowest level of control – a standard (non-hybrid) state-constrained optimal control problem for each two consecutive pairs

$$\left((0, \xi_0),\, (t_s^1, \xi_s^1)^*,\, (t_s^2, \xi_s^2)^*,\, \ldots,\, (t_s^{M^*}, \xi_s^{M^*})^*,\, (T, \xi_T)\right). \tag{5.1}$$

Refer to [7, 17, 21, 22, 26, 30, 53] for more details on the solution of the state-constrained optimization problem.

Compared to the presented three-tier structure, the preliminary bimodal approach in Chapter 4 shows a simpler set-up composed only by two levels of abstraction or, more precisely, consisting exactly in the two lower levels of the above hierarchic arrangement. The additional, superordinate automaton level is not necessary since the geometric framework is clearly given by the problem's definition. This fact was previously explained in the discussions at the beginning of this

chapter. As a consequence, the bimodal solution procedure, which is based on the two-level hierarchic structure and explicitly explained in Section 4.5, represents an easily approachable special case of the general multimodal point-to-point problem. However, the idea of taking advantage of the revealed hierarchic structure in the way that the optimal control problem is partially solved on each level of control can be observed already in the simplified bimodal approach.

In the following, after a precise definition of the regional dynamics system in Section 5.2 and Section 5.3, the different levels of abstraction and their inherent functions are described in greater detail. This is done by explaining, in Section 5.4, the transformation of a given geometric structure into the associated transition automaton and, in Section 5.6, by deriving a Dynamic Programming algorithm representing an applicable solution scheme for the considered point-to-point problem, which is based on both the transition sequences provided by the automaton on the highest level of control and the cost functions $c(\cdot, \cdot, \cdot, \cdot, \cdot)$ computed on the lowest level of control.

5.2 The Multimodal System – Regions and Geometric Framework

A straight-forward extension of the regional definitions and assumptions, introduced in the bimodal case (Section 4.1), yields to a precise description of the multimodal geometric framework and, hence, lays the foundation for the further approach to a solution of the hybrid point-to-point problem in multiple regions.

The descriptive explanations in Section 3.1 already provided an pictorial understanding of what constitutes a general multimodal system. As indicated in Figure 3.1, in a regional dynamics system, the compact state space $\mathcal{X}$ is divided into an arbirtrary, but finite, number q of open, connected, and simply connected regions $D_i,\ i \in \mathcal{I} = \{1, 2, \ldots, q\},\ q \in \mathbb{N}$, such that

$$\mathcal{X} = \bigcup_{i=1}^{q} (D_i \cup \partial D_i), \tag{5.2}$$

where $D_i \cap D_j = \emptyset,\ \forall i, j \in \mathcal{I},\ i \neq j$. As in the bimodal case, the boundaries

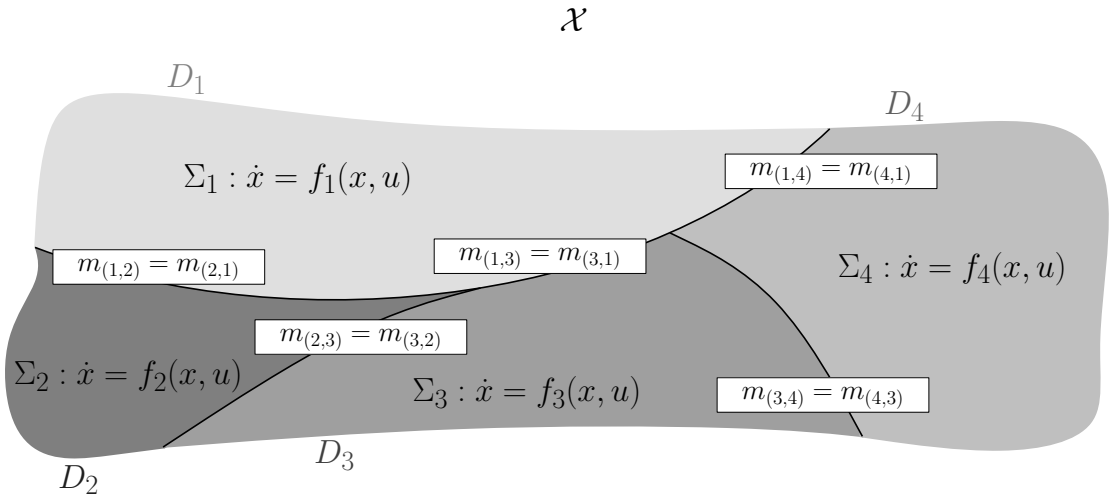


Figure 5.1: System with Regional Dynamics – Boundaries

$\partial D_i,\ i \in \mathcal{I}$ are assumed to be finite unions of closed, smooth codimension one submanifolds s_i^k of $\mathcal{X}$ and are precisely given by equation (4.2). However, in contrast to the bimodal framework, where only *one* switching surface ∂D exists dividing the state space $\mathcal{X}$ into its two regions D_1 and D_2 as depicted in Figure 3.2, the general case shows multiple *switching manifolds* $m_{(i,j)}$ defined by

$$m_{(i,j)} = \partial D_i \cap \partial D_j, \quad i,j \in \mathcal{I}, \tag{5.3}$$

where $m_{(i,j)} \neq \emptyset$ if region i and j are adjacent. This definition is visualized in Figure 5.1 for the regional dynamics example in Figure 3.1. As a consequence of equation (5.3), the relations

$$m_{(i,j)} = m_{(j,i)} \quad \text{and} \quad m_{(i,i)} = \partial D_i, \quad i,j \in \mathcal{I} \tag{5.4}$$

hold. Note that $m_{(i,i)}$ contains the part of the boundary ∂D_i which does not meet the boundary of any other region. Moreover, the meaning of the regional structure is consistent with the one in the bimodal case (Section 4.1); that is, with each region $D_i,\ i \in \mathcal{I}$ a time-invariant vector field $f_i(x,u)$ is associated which uniquely describes the continuous dynamics in the corresponding partition:

$$\dot{x}(t) = f_i(x(t),u(t)) \quad \text{if} \ \ x(t) \in D_i. \tag{5.5}$$

Analogous to Section 4.1, Assumption 4.1 holds for all $i \in \mathcal{I} = \{1,2,\ldots,q\}$.

The previous considerations represent a pure extension of the bimodal case in Section 4.1, where simply a larger set $\mathcal{I} = \{1, 2, \ldots, q\}$ is chosen. However, in the following section, which defines the possible executions in a regional dynamics system and, in particular, the dynamics of the multimodal system on the boundaries, a new feature, the discrete control variable e, is introduced, which precisely specifies the transition behavior of the continuous state arrived at a boundary. As mentioned in the introductory comments at the beginning of this chapter, because of the limited dynamics in the bimodal case, a discrete input variable was not required in the precedent approach of Chapter 4.

5.3 The Multimodal System – Dynamics and Executions

A novel attribute, the discrete input variable e, qualifies the continuous state's behavior at a switching manifold $m_{(i,j)}$, where a decision is to be made between "passing through" the boundary into a new region with a different dynamical regime and "bouncing back" into the previous region, where the state's execution is ruled by the same dynamics again.

Proceeding in a similar manner as in the bimodal case (Section 4.2), the goal of this section is to precisely define what constitutes executions of systems as introduced in Section 5.2; that is, to specify the way the system's continuous state x travels through the different domains $D_i,\ i \in \mathcal{I} = \{1, 2, \ldots, q\}$.

In a first step, the definition of the discrete state space $\mathcal{Q}$, initially introduced for the bimodal case of Section 4.2, is extended to multiple regions. In one-to-one correspondence to the enlarged set of regions $\mathcal{D} = \{D_i :\ i \in \mathcal{I}\}$, the discrete state space of a multimodal system is given by $Q = \{q_i \mid i \in \mathcal{I}\}$. In fact, the discrete state $q(t)$ and the continuous state $x(t)$ are related in the following way: For an interior point $x(t) \in D_i$, the discrete state $q(t)$ is clearly given by $q(t) = q_i$, cf. Section 4.2. However, when the state $x(t)$ lies in a boundary segment $m_{(i,j)},\ i, j \in \mathcal{I}$, the interpretation of the possible discrete state values $q(t-) = q_i$ or $q(t-) = q_j$

is that the continuous state has arrived at $x(t)$ along a trajectory which most recently lay in D_i, or respectively, D_j, and a switch of the discrete state to, say, $q(t) = q_i$, indicates that the system's trajectory will evolve in D_i under the ith vector field on a time interval with initial instant t.

Recalling the notation x_k emphasizing the dynamical regime f_k which determines the execution of the trajectory at a given point $x_k(t) \in D_k \cap \partial D_k$, i.e., $\dot{x}_k(t) = f_k(x_k(t), u(t))$, $k \in \mathcal{I}$, the transition behavior of the multimodal system can be described as follows: Given a continuous starting state $\xi_0 \in D_i$, $i \in \mathcal{I}$ and the associated discrete start state $q_i \in \mathcal{Q}$, the *continuous-time* control input $u(\cdot) \in \mathcal{U}$ gives rise to a trajectory evolving according to $\dot{x}_i = f_i(x_i, u)$. If there is a finite time t_s such that the state enters a *switching manifold*

$$m_{(i,j)} = \partial D_i \cap \partial D_j, \quad j \in \mathcal{I}\,, \tag{5.6}$$

that is,

$$\xi_s = \lim_{t \to t_s} x_i(t) \ \in m_{(i,j)}, \tag{5.7}$$

then, corresponding to the previous explanations and illustrations in Section 3.1 and Figure 3.1, there are two different possibilities of further execution:

(i) One possible option is that the trajectory *passes through* the switching manifold and, in the following, evolves in region j as $\dot{x}_j = f_j(x_j, u)$ with the initial condition $x_j(t_s) = \xi_s$ until a further possible intersection with a switching surface $m_{(j,k)}$, $k \in \mathcal{I}$. In this case, the switching at $m_{(i,j)}$ corresponds to a transition from domain i to j.

(ii) However, it is also counted as a switch, if the continuous state x after arriving at $m_{(i,j)}$ under the dynamics $f_i(x_i, u)$ *bounces back* into region i, where, again, the state's dynamics are determined by $\dot{x}_i = f_i(x_i, u)$ with initial condition $x_i(t_s) = \xi_s$ (until a further possible intersection with a switching manifold $m_{(i,j)}$).

Note that at a switching point $(t_s^i,\, \xi_s^i)$ in the case of "passing through" as well as in the case of "bouncing back" the vector field determining the execution of the hybrid trajectory changes discontinuously.

In brief, characteristic for the evolution in a general regional dynamics system is the ambiguity in the continuous state's behavior when entered a switching manifold $m_{(i,j)},\ i,j \in \mathcal{I}$. The decision on which way will be taken is determined by a *discrete* control input $e \in \mathcal{E}$, where $\mathcal{E}$ denotes a finite set of transition labels. In this context, the discrete value $e \in \mathcal{E}$ uniquely specifies the region of the trajectory's further execution once arrived at a switching manifold $m_{(i,j)},\ i,j \in \mathcal{I}$. Clearly, the set of possible transitions at a given switching point $\xi_s \in m_{(i,j)}$ depends on the location of ξ_s in the state space $\mathcal{X}$ and also on the family of vector fields defined at ξ_s. If, for example, ξ_s lies in $m_{(i,j)} \cap m_{(k,l)},\ i,j,k,l \in \mathcal{I}$, the location of ξ_s only allows a switch to region D_i, D_j, D_k, and D_l. However, supposed there exists no $u \in U$ such that $f_j(\xi_s, u)$ meets the open set D_j, then, the set of possible future regions is to be cut down to D_i, D_k, and D_l.

In the following, the previously suggested *(controlled) discrete dynamics*, inherent in the considered multimodal system behavior, are introduced in a formal way: A controlled discrete transition from $q_i \in \mathcal{Q}$ to $q_j \in \mathcal{Q}$ is defined at the continuous state x and denoted by

$$DSC: \quad q_j = \Gamma\left(q_i, e_{ij}\right), \quad j \neq i, \tag{5.8}$$

in case

(i) $x \in m_{(i,j)}$,

(ii) there exists a continuous control $u \in U$ such that the oriented vector at x given by the vector field $f_j(x,u)$ meets the open set D_j within all neighborhoods of x, and

(iii) the controlled input event $e \in \mathcal{E}$ takes the value e_{ij}.

In addition, a controlled discrete transition from $q_i \in \mathcal{Q}$ back to $q_i \in \mathcal{Q}$ is defined at the continuous state x and denoted by

$$DSC: \quad q_i = \Gamma\left(q_i, e_{ii}\right), \tag{5.9}$$

in case

(i) $x \in m_{(i,j)}, \quad j \in \mathcal{I}$,

(ii) there exists a continuous control $u \in U$ such that the oriented vector at x given by the vector field $f_i(x,u)$ meets the open set D_i within all neighborhoods of x, and

(iii) the discrete control input $e \in \mathcal{E}$ takes the value e_{ii}.

In a nutshell, the above definitions state that, in case the trajectory $x(\cdot)$ enters a switching manifold $m_{(i,j)},\ i,j \in \mathcal{I}$ from domain D_i, one of the discrete inputs, e_{ij} or e_{ii}, identifies the region of further execution, D_j or D_i, respectively. The function $\Gamma : \mathcal{Q} \times \mathbb{R}^n \times \mathcal{E} \to \mathcal{Q}$ represents a time-independent, partially defined *discrete transition map*, which specifies transitions only for points on the boundary $x \in m_{(i,j)},\ i,j \in \mathcal{I}$. The set $\mathcal{E}$ contains the labels of all possible transitions in the given partitioned state space $\mathcal{X}$. Later, in Section 5.4, an untimed automaton, a discrete representation of the multiregional state space $\mathcal{X}$, is introduced, which explicitly specifies the desired set $\mathcal{E}$.

In addition, rather more significant is the following assumption: If at any $x \in m_{(i,j)},\ i,j \in \mathcal{I}$ one of the two conditions above, (5.8) or (5.9), is satisfied, then, the condition holds for all other continuous states in $m_{(i,j)}$, too. Hence, the discrete transition events $e \in \mathcal{E}$ are well defined.

Moreover, note that the discrete dynamics, (5.8) and (5.9), with inputs $e_{ij},\ i,j \in \mathcal{I}$ represent the continuous state's switches between different regions and cause the discontinuous behavior in the discrete state's trajectory $q(\cdot)$. Consequently, in accordance with the defining comments at the beginning of this section, the

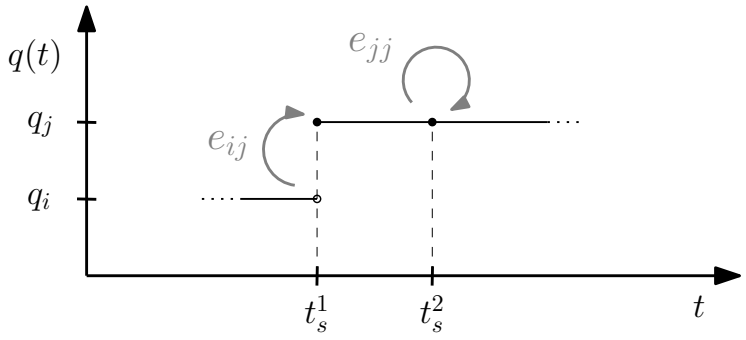


Figure 5.2: The Discrete Input Values e_{ij} and e_{jj} Qualify the Discontinuous Transitions of the Discrete-State Trajectory $q(\cdot)$.

discrete-valued state $q(\cdot)$ has right-continuous trajectories in $\mathbb{R}$, which are piecewise constant. These characteristics regarding the discrete state q were previously illustrated for the bimodal case in Section 4.2 and, especially, in Figure 4.1. Additionally, Figure 5.2 visualizes the impact of the controlled discrete transitions, (5.8) and (5.9), on the discrete variable $q(\cdot)$ at two specific time instances t_s^1 and t_s^2, where $x(t_s^1) \in m_{(i,j)},\ i,j \in \mathcal{I}$ and $x(t_s^2) \in m_{(j,k)},\ k \in \mathcal{I}$.

In conclusion, a hybrid execution is uniquely determined by a given initial condition $\xi_0 \in D_i,\ i \in \mathcal{I}$, the continuous-time control input $u(\cdot) \in \mathcal{U}$, and a *compatible* discrete control sequence

$$S(\overline{\tau}, w) = \left(\left(t_s^1,\, e_1\right),\, \left(t_s^2,\, e_2\right),\, \ldots,\, \left(t_s^M,\, e_M\right)\right), \tag{5.10}$$

where $\overline{\tau} = (t_s^1,\, t_s^2,\, \ldots,\, t_s^M),\ 0 < t_s^1 < t_s^2 < \cdots < t_s^M < T$ denotes the strictly increasing sequence of switching times and $w = e_1 e_2 \ldots e_M$ represents the corresponding string of appropriate discrete control inputs $e_i \in \mathcal{E},\ i \in \{1, 2, \ldots, M\}$. The positive integer M determines the number of switches. Of course, the sequence $S(\overline{\tau}, w)$ is empty, if no switch occurs along the hybrid trajectory. In brief, the admissible control actions available are the continuous control $u(\cdot) \in \mathcal{U}$ and the discrete control input sequence $S(\overline{\tau}, w)$. Both control inputs are part of our optimization problem which is explicitly defined in Section 5.5. In fact, not only is the system's input hybrid in nature, but so is the description of the actual execution consisting of the continuous state $x(\cdot)$ and the discrete variable $q(\cdot)$. Moreover, from these specifications of the hybrid system executions, it follows

that over any finite interval $[0, T]$, starting at $\xi_0 \in D_i$ for some $i \in \mathcal{I}$, the resulting controlled trajectory is continuous for any continuous control function $u(\cdot) \in \mathcal{U}$ and any discrete control input sequence $S(\overline{\tau}, w)$. An example for the resulting controlled dynamics of a hybrid trajectory evolving in a partitioned state space $\mathcal{X}$ is pictured in Figure 3.1.

Remark 5.1. Important to note is that most of the former approaches, such as [59] and [68], define the transition behavior of their hybrid dynamic systems in a different way. In contrast to the multimodal system introduced in the precedent sections, the models in [59] and [68] do not include the possibility of "bouncing back" at a switching manifold. However, since this work's multimodal system allows to inhibit the "bouncing back" switches (and also other kinds of switches) on a higher level of control, cf. Section 5.4, which may yield to the models proposed in [59] and [68], the recently presented hybrid modeling turns out to be a more general approach in view of possible ways of transition. □

The subsequent section introduces the previously announced transition automaton, which relates the geometric structure of the state space $\mathcal{X}$ to the set of possible transitions $\mathcal{E}$ and, thus, specifies the discrete dynamics (5.8) and (5.9). In doing so, the automaton encodes the manner in which the system's trajectory can transition from one region to another.

5.4 The Transition Automaton

A key concept in the hierarchic approach to a solution of the multimodal point-to-point problem, cf. Section 5.1, is the mapping of the given multiregional state space $\mathcal{X}$ to a finite state machine, referred to as the transition automaton, which contains information about the contributing regions and their connections among each other.

In this section, an important part of the multimodal solution scheme or, more precisely, the highest level of abstraction in the hierarchical structure introduced

in Section 5.1, is established, distinguishing the general multiregional approach from the preliminary bimodal considerations in Chapter 4.

To begin with, the hybrid optimal control problem considered in this work and previously illustrated in Section 3.1 is briefly stated again: Given a specific cost function, the goal is to determine the optimal path of going from a given initial state $\xi_0 = x_{i_0}(0) \in D_{i_0}$ to a fixed final state $\xi_T \in D_{i_T}$ during a time horizon T, where T is also specified a priori. In order to solve this optimization problem and even in order to formulate this problem in a precise manner, what is done in Section 5.5, a discrete representation of the geometric framework, explicitly specifying the connections between the constituent regions $D_i,\ i \in \mathcal{I}$, is indispensable. Therefore, a *deterministic automaton*, the transition automaton, is introduced modeling the switching surfaces $m_{(i,j)},\ i,j \in \mathcal{I}$ in the way that a transition between the *discrete* automaton states i and j, representing the region D_i and D_j, respectively, is only possible if the switching surface $m_{(i,j)} \neq \emptyset$. Furthermore, information about the optimal control problem, in particular, about the initial region D_{i_0}, $\xi_0 \in D_{i_0}$ and the final region D_{i_T}, $\xi_T \in D_{i_T}$ are incorporated in the automaton. Consequently, the automaton answers the question: Which ways, i.e. which sequences of transitions, are possible in order to get from ξ_0 to ξ_T?

A formal definition of the transition automaton, following the notation of [16], is given by the six-tupel

$$A = (\mathcal{D}_s, \mathcal{E}, g, h, d_0, \mathcal{D}_m), \tag{5.11}$$

where

- the set of (automaton) states is $\mathcal{D}_s = \mathcal{I}$,
- the set of events is given by $\mathcal{E} = \{e_{ij} \mid i,j \in \mathcal{I}\}$,
- the transition function is defined as $g(i, e_{ij}) = j$ if $m_{(i,j)} \neq \emptyset,\ i,j \in \mathcal{I}$ and is not defined for all other cases,
- the initial state $d_0 = i_0$, where $\xi_0 \in D_{i_0}$, and

- the set of marked states is given by $\mathcal{D}_m = \{i_T \mid \xi_T \in D_{i_T}\}$.

The active event function h is defined by the specifications above.

In this formalism, the states $\mathcal{D}_s = \mathcal{I}$ represent the discrete state space $\mathcal{Q} = \{q_i \mid i \in \mathcal{I}\}$, and, with this, the different regions D_i, $i \in \mathcal{I}$. An event e_{ij} characterizes the transition, i.e. the switch, from region D_i to region D_j and the set of events $\mathcal{E}$ corresponds to the previously defined set of transition labels, cf. Section 5.3. Comparable with the discrete transition map Γ in Section 5.3, the transition function g indicates, on the automaton level, which transitions $e_{ij} \in \mathcal{E}$ are possible for a given state $i \in \mathcal{D}_s$. In addition, especially noteworthy is that the elements $\mathcal{D}_s$, $\mathcal{E}$, g, and h of the six-tupel (5.11), specifying the transition automaton A, are immediately given by the geometric structure of the state space $\mathcal{X}$, whereas the initial state of the automaton d_0 and the set of marked states $\mathcal{D}_s$ depend on the HOCP formulation, namely on the initial and final state, ξ_0 and ξ_T, respectively.

An implicit assumption in the above model of the regional dynamics system, (5.11), is that the condition (ii) in the definition of the discrete dynamics, (5.8) and (5.9), does not restrict the set of regions the continuous state x can go to, when arrived at a switching manifold $m_{(i,j)}$, $i, j \in \mathcal{I}$. That is, it is taken for granted that there always exist (continuous) control values $u_i, u_j \in U$ such that the oriented vectors at $x \in m_{(i,j)}$, $i, j \in \mathcal{I}$ given by the $f_i(x, u_i)$ and $f_j(x, u_j)$ meet the open set D_i and D_j,respectively, within all neighborhoods of x.

Note that in the automaton description (5.11), the geometric shape of a region $D_i, i \in \mathcal{I}$ is reduced to a discrete state i and the switching manifold $m_{(i,j)}$, $i, j \in \mathcal{I}$ to a single event e_{ij}. Furthermore, an important observation is that the automaton does contain information about the initial state $\xi_0 \in D_{i_0}$ and the final state $\xi_T \in D_{i_T}$, but does not contain any time-valued information.

A simple example for the transformation of a given geometric structure to the corresponding transition automaton is depicted in Figure 5.3, where a direct graph

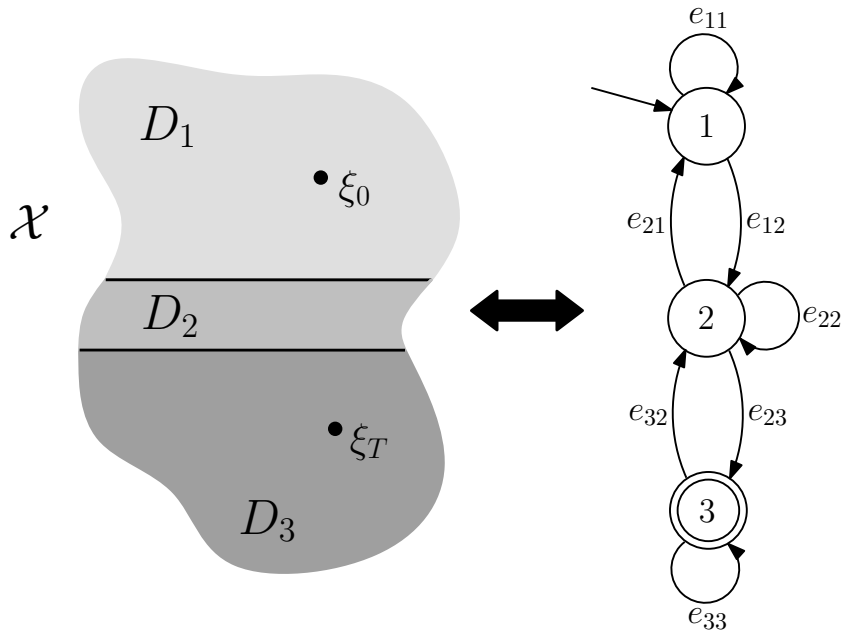


Figure 5.3: Transformation of a Given Geometric Structure to the Corresponding Transition Automaton

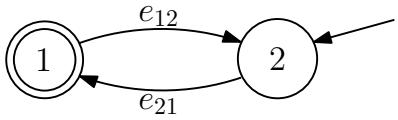


Figure 5.4: Transition Automaton Associated with the Bimodal Point-To-Point Problem of Chapter 4; $i_0 = 2$ and $i_T = 1$

representation, as proposed in [16], is used to picture the automaton. In this illustration, arrows between different states imply the continuous state's transition from one region to another, whereas self-loops signalize the "bouncing back" behavior of the continuous state x.The transition automaton associated with the initially considered bimodal point-to-point problem, defined in Section 4.3, is shown in Figure 5.4, where, because of the limited dynamics, cf. Section 4.2, self-loops does not appear. In fact, depending on the initial and final region, the bimodal case allows only two different automaton representations characterized by $i_0 = i_T$ or $i_0 \neq i_T$. The illustration in Figure 5.4 assumes $i_0 = 2$ and $i_T = 1$ for the initial and final region.

Remark 5.2. The previously introduced transition automaton, a discrete model of the multiregional state space $\mathcal{X}$, represents the connections between the constituent regions $D_i,\ i \in \mathcal{I}$ and displays possible sequences of transitions leading

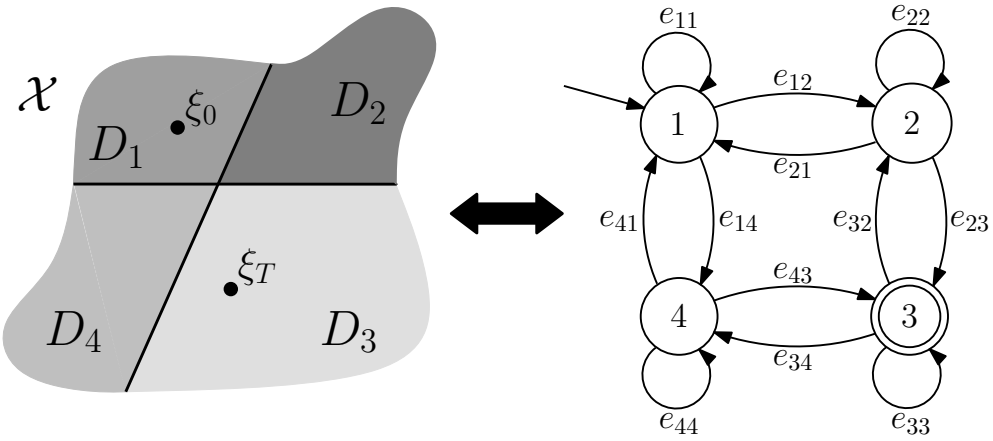


Figure 5.5: Example For The Incorporation of Additional Switching Rules on the Automaton Level

from i_0, where $\xi_0 \in D_{i_0}$, to i_T : $\xi_T \in D_{i_T}$, $i_0, i_T \in \mathcal{I}$. However, especially interesting is that, on this highest level of abstraction, arbitrary switching rules can be incorporated. An example is given in Figure 5.5, where a state space $\mathcal{X}$ composed of four different regions is considered. In this case, the transitions from state 1 to 3 and from state 2 to 4, that is, switches from region D_1 to D_3 and from D_2 to D_4, are obviated by the proposed automaton. Moreover, in order to disable undesirable transitions, supervisory control [16] and other related methods can be applied on this level of control. Note that the transversality assumptions on the vector fields $f_i(x,u)$, $i \in \mathcal{I}$, Assumption 4.1, can be partially relaxed if certain transitions e_{ij}, $i,j \in \mathcal{I}$ are inhibited by the automaton. □

A crucial role in the further approach to a hybrid optimal solution of the multimodal point-to-point problem, illustrated in Section 3.1, plays the language $L(A)$ generated by the transition automaton A defined in (5.11), cf. [16],

$$L(A) = \left\{ e \in \mathcal{E}^* \;\middle|\; g(d_0, e) \text{ is defined} \right\}, \tag{5.12}$$

where $\mathcal{E}^*$ denotes the set of all finite strings, or sequences, of elements in $\mathcal{E}$ including the empty string ϵ and $g(d_0, e)$, $e \in \mathcal{E}^*$ is interpreted in the following recursive manner, cf. [16]:

$$g(i, \epsilon) \quad = \quad i, \quad i \in \mathcal{D}_s, \tag{5.13}$$

$$g(i, uv) \quad = \quad g\left(g\left(x, u\right), v\right) \quad \text{for } u \in \mathcal{E}^* \text{ and } v \in \mathcal{E}. \tag{5.14}$$

The notation uv indicates the concatenation of string u and v. In particular, the marked language $L_m(A)$ associated with the automaton A given by (5.11),

$$L_m(A) = \left\{ e \in L(A) \,\middle|\, g(d_0, e) \in \mathcal{D}_m \right\}, \tag{5.15}$$

is used to formulate and solve the multimodal point-to-point problem. In the considered framework, the marked language $L_m(A)$ represents the set of *all strings*, that is, of all sequences of transitions, leading from state i_0, where $\xi_0 \in D_{i_0}$, to state i_T with $\xi_T \in D_{i_T}$. Refer to [16] for a more detailed introduction to the languages generated by a deterministic automaton.

With the objective of restricting the strings $w \in L_m(A)$ to a certain length M, i.e., to find all ways from ξ_0 to ξ_T using exactly M switches, an additional language L_M is defined as

$$L_M = \left\{ e \in \mathcal{E}^* \,\middle|\, |e| = M \right\}, \tag{5.16}$$

where the function $|\cdot| : \mathcal{E}^* \to \mathbb{N} \cup \{0\}$ determines the length of a string $e \in \mathcal{E}^*$, that is, the number of events the string e consists of. The length of the empty string ϵ is zero, $|\epsilon| = 0$. Finally, the intersection

$$L_m(M, A) = L_m(A) \cap L_M \tag{5.17}$$

provides the desired set of strings leading from state i_0 to i_T and consisting of *exactly M events*. Moreover, especially useful in the later derivation of the general Hybrid Bellman Equation for multimodal systems (Section 5.6), is the definition of the language

$$F_K(M, A) = L_K \;\cap\; \mathit{suff}(L_m(M, A)) \tag{5.18}$$

as the set of all suffices of $L(M, A)$ consisting of K events, where the set $\mathit{suff}(L) = \{s \in \mathcal{E}^* \mid \exists\, u \in \mathcal{E}^* \text{ with } us \in L\}$ represents the suffix closure of the language L. Note that the set $F_0(M, A)$ only contains the empty string ϵ, i.e., $F_0(M, A) = \{\epsilon\}$.

In conclusion, the language $L_m(M, A)$, (5.17), associated with the introduced automaton A, (5.11), provides *global accessibility relations* in the way that words

$w = e_{i_0 j_1} e_{i_2 j_2} e_{i_3 j_3} \ldots e_{i_M i_T} \in L_m(A)$ with a prescribed number M of events are determined which correspond to potential accessibility relations, namely possible sequences of switchings, between the specified discrete states q_{i_0}, indicating that $\xi_0 \in D_{i_0}$, and q_{i_T}, where $\xi_T \in D_{i_T}$, along trajectories with a prescribed number M of switchings. At the base (continuous) system level, Section 5.6, these global accessibility relations are used to constrain the infimization in the Hybrid Bellman Equation, that is, in the characterization of the optimal switching states $(\xi_s^i)^*$, $i \in \{1, 2, \ldots, M\}$, the optimal discrete control sequence, $S^*(\overline{\tau}, w)$, and the associated optimal continuous control functions. The *global knowledge* available through the automaton A, (5.11), and even the model of the automaton itself are solely useful and necessary since, in contrast to other hybrid optimal control approaches, this work considers an optimization problem, where both, the initial and the final state of the hybrid optimal trajectory, are given *a priori*, see Section 3.1.

Remark 5.3. In the bimodal case, Chapter 4, which might be represented by an automaton as depicted in Figure 5.4, the language $L_m(M, A)$, $M \in \mathbb{N} \cup \{0\}$ contains exactly one word, i.e., one possible sequence of switchings, if M is even and $i_0 = i_T$ or M is odd and $i_0 \neq i_T$; otherwise, the set $L_m(M, A)$ is empty. As a consequence, depending on i_0 and i_T, only odd or even numbers M have to be taken into account and the uniquely given sequence of switchings is directly incorporated in the Hybrid Bellman Equation (4.7). □

Against this background, the following section precisely defines the point-to-point problem for a general regional dynamics system.

5.5 The Optimization Problem Formulation

A mathematically precise formulation of the considered hybrid optimal control problem in multiple regions involves, compared to the bimodal case (Section 4.3), two important additional concepts: the definition of a discrete control input sequence $S(\overline{\tau}, w)$, given by (5.10), and the global accessibility relations provided by the language $L_m(M, A)$, equation (5.17).

Given an *upper bound $N \in \mathbb{N}_0$ on the total number of switches*, the multimodal point-to-point problem, previously illustrated in Section 3.1, is formally defined by the following problem statement:

Problem 5.1 (The Multimodal Point-To-Point Problem).

$$\mathcal{P}_N : \inf_{\substack{u(\cdot)\in\mathcal{U},\\ S(\overline{\tau},w)}} \int_0^T \ell(x(t),u(t))\,dt \tag{5.19}$$

subject to, for $0 \leq M \leq N$,

- *the geometric structure*

$$w = e_{i_1 j_1} e_{i_2 j_2} \dots e_{i_M j_M} \in L_m(M,A), \tag{5.20}$$

- *the discrete dynamics*

$$q_{j_k} = \Gamma\ (q_{i_k}, e_{i_k j_k}) \tag{5.21}$$

  at a switching time t_s^k, where $1 \leq k \leq M$, yielding to the discrete-state dynamics $q(t) = q_{i_k},\ t \in \left[t_s^{k-1},\, t_s^k\right)$ with $0 < k \leq M+1$ and $i_{M+1} = j_M$,

- *the continuous-state dynamics*

$$\dot{x}_{i_k}(t) = f_{i_k}(x_{i_k}(t), u(t)), \quad t \in \left[t_s^{k-1},\, t_s^k\right), \tag{5.22}$$

  where $0 < k \leq M+1$ and $i_{M+1} = j_M$,

- *and the corresponding initial and final conditions*

$$\begin{aligned} x(0) &= x_{i_1}(t_s^0) &&= \xi_0 &&\in D_{i_0}, \\ x_{i_{k+1}}(t_s^k) &= \lim_{t\to t_s^k} x_{i_k}(t) &&= \xi_s^k &&\in m_{(i_k,j_k)}, \\ x(T) &= x_{i_{M+1}}(t_s^{M+1}) &&= \xi_T &&\in D_{i_T}, \end{aligned} \tag{5.23}$$

  if $0 < k \leq M$.

□

Note that $\xi_0, \xi_T \notin \partial D_i,\ i \in \mathcal{I}$ and $(t_s^k,\, \xi_s^k),\ 0 < k \leq M$ denotes the kth switching point along the trajectory.

Based on above's precise problem definition, the subsequent section finally answers the original question of this thesis, cf. Section 3.1, "*Which path is optimal* if the system's continuous state is required to pass, under a regional dynamics regime, from a given starting point to a fixed final state during an *a priori* specified time horizon?".

5.6 The Hybrid Bellman Equation

The main result of this work, the Hybrid Bellman Equation for multimodal systems, is presented.

In fact, the multiregional point-to-point problem, precisely stated in Section 5.5, provided the motivation for all previous considerations: Starting with a brief review on the basic optimal control concepts in Chapter 2, especially illustrating the Dynamic Programming idea fundamental to the actual hybrid solution procedure, the point-to-point problem in multiple regions is explained in an informal, illustrative way, Section 3.1, and a rough sketch of the proposed solution scheme is shown in Section 3.2. The following mathematically precise bimodal approach, Chapter 4, represents a great basis for general multiregional approach begun in Section 5.2 and Section 5.3 by a clear definition of the multiregional geometry and dynamics and the introduction of a discrete representation of the regional geometric framework in Section 5.4. This section presents the final conclusion of all precedent results by solving the point-to-point problem in multiple regions and, indeed, providing a recursive algorithm providing a characterization of global optimality.

In order to derive a Dynamic Programming recursion describing the cost-to-go dynamics of the general multimodal point-to-point problem, let $end\colon\ \mathcal{E}^* \setminus \{\epsilon\} \ \rightarrow \mathcal{I}$ denote the mapping

$$end(s) = end(e_{i_1 j_1} e_{i_2 j_2} \dots e_{i_k j_k}) = j_k, \tag{5.24}$$

which is used to specify the region D_{j_k} associated with the last index of the last transition label $e_{i_k j_k}$ of a string s. Additionally, as in the bimodal case, a concise formulation of the multimodal Hybrid Bellman Equation is based on the cost-to-go function $V^{\cdot}(\cdot,\,\cdot,\,\cdot,\,\cdot,\,\cdot)$, see Definition 4.2, and the cost $c(\cdot,\,\cdot,\,\cdot,\,\cdot,\,\cdot)$, given by Definition 4.1, where $\mathcal{I} = \{1, 2,\,\ldots,\,q\}$ in the multiregional case. However, of particular importance is that, in the multimodal case, the determination of the functions $V^M(\xi_1,\,q_{i_1},\,\xi_2,\,q_{i_2},\,\tau)$ is based on global accessibility relations represented by a particular transition automaton $\widetilde{A}$. That is, given a certain geometric framework, specifying the variables $\mathcal{D}_s$, $\mathcal{E}$, g, and h in the definition of the transition automaton $\widetilde{A}$, cf. (5.11), in order to obtain the value $V^M(\xi_1,\,q_{i_1},\,\xi_2,\,q_{i_2},\,\tau)$ only hybrid trajectories have to taken into account, whose execution is determined by a discrete input word, i.e., by a switching sequence, satisfying $w \in L_m(M, \widetilde{A}) = L_m(\widetilde{A}) \cap L_M$, where the corresponding automaton $\widetilde{A}$ is given by $\widetilde{A} = (\mathcal{D}_s, \mathcal{E}, g, h, i_1, \{i_2\})$, cf. (5.11), and the languages $L_m(\widetilde{A})$ and L_M are defined as in (5.15) and (5.16), respectively.

Finally, the main theorem of this work, a solution scheme for the general multiregional point-to-point problem (Problem 5.1), reads as follows:

Theorem 5.1 (The Hybrid Bellman Equation for Multimodal Systems). *Assume that all hypotheses for the existence and uniqueness of regional dynamics hybrid systems hold and that all infima exist in the definition of the hybrid value functions $V^{\cdot}(\cdot,\cdot,\cdot,\cdot,\cdot)$, for all admissible argument values, whenever the expressions are finite. Then, given the transition automaton A, cf.* (5.11)*, corresponding to the multimodal point-to-point problem, Problem 5.1, the recurrence relation is expressed by*

$$
\begin{aligned}
&V^K(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau) \\
&= \inf_{t\in(0,\tau)} \inf_{\xi\in m_{(i_1,j)}} \inf_{j\in\mathcal{I}} \Big\{ c(\xi_1, q_{i_1}, \xi, q_j, t) + V^{K-1}(\xi, q_j, \xi_2, q_{i_2}, \tau - t) \Big\} \qquad (5.25)
\end{aligned}
$$

such that

$$e \;=\; e_{i_1 j}\,, \qquad w \;\in\; F_{K-1}(M,A)\,, \tag{5.26}$$

$$end(w) \;=\; i_2\,, \qquad ew \;\in\; F_K(M,A)\,. \tag{5.27}$$

This relation holds for $0 < K \leq M$. *The initial condition of the recursive scheme is given by*

$$V^0(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau) \;=\; c(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau). \tag{5.28}$$

□

Since the multimodal point-to-point problem (Problem 5.1) does not insist on an *a priori* given number of switches $M,\ 0 \leq M \leq N$, where N denotes the given upper bound on the total number of switches along a hybrid trajectory, in a last step, the functions $V^M(\xi_1, q_{i_1}, \xi_2, q_{i_2}, T)$ are related to the original problem. Recalling that $\xi_0 \in D_{i_0}$ and $\xi_T \in D_{i_T},\ i_0, i_T \in \mathcal{I}$, the optimal cost associated with the original problem $W^N(\xi_0, q_{i_0}, \xi_T, q_{i_T}, T)$ is given by (4.9)

$$W^N(\xi_0, q_{i_0}, \xi_T, q_{i_T}, T) = \min_{0 \leq K \leq N} V^K(\xi_0, q_{i_0}, \xi_T, q_{i_T}, T)\,. \tag{5.29}$$

By carefully studying the above recursion relation (5.25) and the associated high level language constraints (5.26) and (5.27), the following notable interpretations are made: As mentioned in Section 5.4, the languages $F_K(M,A),\ 0 < K \leq N$, which represent the global accessibility conditions in a given geometric framework, may enormously – depending on the structure of the partitioned state space $\mathcal{X}$ and the upper bound N – constrain the effort needed to accomplish the recursion proposed in Theorem 5.1. Precisely speaking, the language $F_K(M,A)$ explicitly specifies which combinations $(K,\, q_{i_1},\, q_{i_2})\,,\ q_{i_1}, q_{i_2} \in \mathcal{I}$ for $V^K(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau),\ 0 < K \leq N$ are relevant and have to be computed. To illustrate this, the geometric setup shown in Figure 5.5 is considered. Assuming the upper bound $N = 3$ on the total number of switches along a hybrid trajectory, in this example, for instance, the value functions $V^K(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau)$ with $q_{i_2} \neq 3$ or $V^2(\xi_1, 3, \xi_2, 3, \tau)$ are not contained in $F_K(M,A)$ and $F_2(M,A)$, respectively,

and can, therefore, be ignored while solving the corresponding hybrid point-to-point problem, Problem 5.1. Furthermore, notice the great analogy between the general multiregional result, Theorem 5.1, and the preliminary bimodal approach, Theorem 4.1, where in both cases the geometric structure restricts the infimization. However, in the bimodal case, these constraints are directly included in the recursion equation (4.7) by the introduced complementary indicator $(\cdot)^c$, whereas in the above Theorem 5.1 the additional equations (5.26) and (5.27) are stated.

Remark 5.4. The regional dynamics system defined in Section 5.2 and Section 5.3 perfectly fits in the general framework presented in [11], where a unified mathematical description of a controlled hybrid dynamical system is introduced. More precisely, this work's model presents a special case of the notions in [11] considering only the possibility of *autonomous switchings*. However, the idea of deriving a Hybrid Bellman Equation based on global accessibility relations provided by an automaton, which represents the structure of the partitioned state space $\mathcal{X}$, can be extended to more general models as, for example, those in [11]. Nevertheless, these extensions and other possible generalizations, cf. Chapter 6, are left to a future endeavor. □

The subsequent section addresses the practical implementation of the theoretical results, concluded by Theorem 5.1, and benefits in its approach highly from the previous considerations on the numerical solution of the bimodal problem, Section 4.5.

5.7 Implementation and Computational Issues

Again, the hierarchic organization of the problem, cf. Section 5.1, turns out to be a helpful structure indicating, in this context, the numerical way of proceeding, where the computational effort needed to accomplish the recursive algorithm, (5.25), is, in particular, influenced by the geometric framework of the regional dynamics system and the upper bound N on the total number of switches along a hybrid trajectory.

The following comments on the computational procedure solving the general multiregional point-to-point problem are mainly drawn from the observations previously made in the bimodal case, Section 4.5, which definitely represents a more concrete and easier to handle problem formulation. As already noticed there, the hierarchical structure of the point-to-point problem not only divides the complex question of finding the optimal hybrid trajectory into different levels of control but, from a numerical point of view, also provides subproblems which can be solved successively as illustrated in the following:

(i) First and additional to the computational steps shown in the bimodal case, the global accessibility relations, namely the languages $F_K(M, A)$ for $0 < K \leq M$ and $0 < M \leq N$, are determined based on the transition automaton A defined in (5.11). Hence, at the beginning of the numerical approach, the subproblem corresponding to the highest level of abstraction is solved. The effort associated with the calculation of these $(N+1)N/2$ languages depends, obviously, on the upper bound N. However, even more important but difficult to estimate is the influence of the state space's geometric structure, characterized by the number of constituent regions and their given composition. In fact, a precise analysis of the relation between the structure of the automaton and the computational effort associated with providing the sets $F_K(M, A)$ is an interesting field for future studies.

(ii) The second preliminary computation to be accomplished before performing the actual Dynamic Programming recursion (5.25) lies in the lowest level of abstraction and prepares, in analogy to the bimodal case in Section 4.5, the functions $c(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau)$ between all reachable pairs of boundary points for all $\tau \in [0, T]$. Indeed, with the results of (i) the functions $c(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau)$ to be calculated can be explicitly stated: If there exists $w = \ldots e_{ij}e_{jk}\cdots \in L_m(M, A),\ 0 < M \leq N$, where $i, j, k \in \mathcal{I}$, then $c(\xi_1, q_i, \xi_2, q_k, \tau)$ with $\xi_1 \in m_{(i,j)}$ and $\xi_2 \in m_{(j,k)}$ must be calculated for all $\tau \in [0, T]$. As a consequence, in the numerical calculations, besides the time

interval $[0, T]$, only boundaries $m_{(i,j)},\ i, j \in \mathcal{I}$ must be discretized, if a word $w \in L_m(M, A),\ 0 < M \leq N$ exists which contains an event $e \in \{e_{ij}, e_{ii}, e_{jj}\}$. The mentioned structural dependence inherent in this computational step makes an exact estimation of the necessary effort impossible, especially, since the surface area of the individual switching manifolds $m_{(i,j)},\ i, j \in \mathcal{I}$ varies and for each manifold can be discretized with a different step size. However, what can be stated as a result of the previous bimodal studies is that the curse of dimensionality, describing the *exponential* growth of the computational effort when increasing the dimensionality n of the state space $\mathcal{X}$, is also inherent in the general multiregional calculations of the non-hybrid state-constrained optimization problems $c(\xi_1,\, q_{i_1},\, \xi_2,\, q_{i_2},\, \tau)$.

(iii) Finally, the recursive equation (5.25) concludes all preliminary results and provides the optimal hybrid trajectory in terms of the optimal discrete control sequence, $S^*\,(\overline{\tau}, w)$, and the optimal switching points and times, $\left(\xi_s^k\right)^*$ and $\left(t_s^k\right)^*$, respectively. Analyzing equation (5.25), the dependence on the geometric structure of the state space $\mathcal{X}$ is observed, appearing in the language constraints, (5.26) and (5.27), and the additional infimization $\inf_{j\in\mathcal{I}}\{\cdot\}$ in (5.25). As mentioned before, also on this level of abstraction, the huge variety of possible geometries and their different properties in view of the considered recursion relation (5.25) turns the estimation of the computational effort into a challenging assignment. However, drawing the parallels between the general relation (5.25) and the bimodal recursion (4.7), it can be noticed that the curse of dimensionality also rules the multiregional Dynamic Programming algorithm (5.25).

In brief, on all levels of control, the geometric framework of the considered regional dynamics system, cf. Section 5.2 and Section 5.3, has a huge influence on the complexity of the underlying computations. However, the precise dependence of the numerical effort on the regional structure, featured by the number of constituent domains $D_i,\ i \in \{1, 2, \ldots, q\}$, the feasible transitions e_{ij} given by

(5.11), the regions of the initial and final state, D_{i_0} and D_{i_T}, respectively, and the upper bound N on the total number of switches along a hybrid trajectory, is fairly complex and represents a fascinating topic for further studies. Nevertheless, important to note is that the idea of integrating the global accessibility relations, provided by the automaton (5.11), into the Dynamic Programming recursion definitely reduces the computational effort in the way that it constrains the infimization in the proposed Hybrid Bellman equation (5.25) and, therefore, also the number of costs $c(\cdot, \cdot, \cdot, \cdot, \cdot)$ to be calculated. Additionally, it can be stated that, in line with the bimodal case, the curse of dimensionality is inherent in both, the computation of the costs $c(\cdot, \cdot, \cdot, \cdot, \cdot)$ and the recursive scheme (5.25), what may result in time-consuming calculations for high-dimensional systems as observed in the multi-agent problem of Chapter 7. Furthermore, the previously, in Section 4.5, highlighted numerical effects arising out of the (computationally necessary) time and boundary discretization apply to the above proposed general solution procedure, too.

Before continuing with a number of examples showing the dynamic behavior of regional dynamics systems subjected to an optimal control policy, a few words are said about the methods and assumptions which finally lead to the numerical results presented in this work. In order to keep the numeric computation simple and focused on the derived recursive relation, the Hybrid Bellman equation (5.25), Assumption 4.2, stated in the bimodal case, is extended in order to fit into the multiregional framework:

Assumption 5.1. *Along a hybrid optimal trajectory $x^*(\cdot)$ associated with a particular multimodal point-to-point problem, Problem 5.1, the individual contributing segments between two consecutive points $\left(t^k, x^k\right)$, $\left(t^{k+1}, x^{k+1}\right)$, $0 \leq k \leq M^*$ of the sequence*

$$\begin{aligned}&\big((0, \xi_0),\, (t_s^1, \xi_s^1)^*,\, (t_s^2, \xi_s^2)^*,\, \ldots,\, (t_s^{M^*}, \xi_s^{M^*})^*,\, (T, \xi_T)\big)\\ &\qquad = \big((t^0, x^0),\, (t^1, x^1),\, (t^2, x^2),\, \ldots,\, (t^{M^*}, x^{M^*}),\, (t^{M^*+1}, x^{M^*+1})\big)\,, \quad (5.30)\end{aligned}$$

where $(t_s^m, \xi_s^m)^* \in (0,T)\times m_{(i,j)},\ m \in \{1, 2, \ldots, M^*\},\ i,j \in \mathcal{I}$ *represents the time-state pair of the* m*th switch along the optimal hybrid trajectory and* M^* *denotes the optimal number of switches, coincide with the trajectories obtained by solving the corresponding unconstrained optimal control problem between the respective states* $\left(t^k, x^k\right)$ *and* $\left(t^{k+1}, x^{k+1}\right)$*:*

$$\mathcal{P} : \inf_{u\in\mathcal{U}} \int_{t^k}^{t^{k+1}} \ell(x(t), u(t))\, dt \tag{5.31}$$

subject to $\dot{x}(t) = f_i\left(x\left(t\right), u\left(t\right)\right)$ *with the initial condition* $x(t^k) = x^k$ *and the final condition* $x(t^{k+1}) = x^{k+1}$*, where* $i \in \mathcal{I}$ *represents the region associated with the original segment of the hybrid optimal trajectory* $x^*(\cdot)$*. More precisely, with the optimal discrete control input* $w^* = e_{i_1 j_1} e_{i_2 j_2} \ldots e_{i_{M^*} j_{M^*}}$*, the governing dynamical regime is given by* $i = i_{k+1}$*.* □

Now, in analogy to the comments in Section 4.5, given a regional dynamics system satisfying Assumption 5.1, the functions $c(\cdot, \cdot, \cdot, \cdot, \cdot)$ preparing the Dynamic Programming algorithm can be considered in an ordinary *unconstrained* optimal control framework. And, again, by restricting the examples to multimodal point-to-point problems characterized by linear dynamics in each constituent region $D_i,\ i \in \mathcal{I}$ and a quadratic overall cost function J, equation (2.40) yield to the desired costs $c(\cdot, \cdot, \cdot, \cdot, \cdot)$.

This chapter is concluded by presenting a few examples highlighting the operation of the derived Hybrid Bellman equation (5.25).

5.8 Examples

Carefully selected, the following examples allow to calculate the cost functions $c(\cdot, \cdot, \cdot, \cdot, \cdot)$ by the simplifying equation (2.40) – that is, Assumption 5.1 holds (cf. Section 5.7) – and illustrate, in particular, the additional features inherent in the general multiregional approach but neglected in the previous binodal considerations (Chapter 4); namely, the ambiguous behavior at a switching manifold and the global accessibility conditions, provided by the transition automaton (5.11) and integrated in the solution procedure, cf. (5.25).

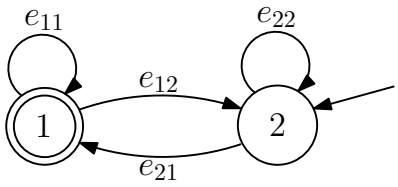


Figure 5.6: Transition Automaton Corresponding To Example 5.1

Consisting only of two regions, the regional dynamics system, considered at first, differs, nevertheless, significantly from the bimodal examples (Section 4.6) in that the dynamical behavior is described by the general definitions in Section 5.3 and, hence, makes a "bounce back" at a switching manifold possible.

Example 5.1. In this example, the planar state space $\mathcal{X} \subset \mathbb{R}^2$ is divided into the two regions $D_1 = \{x \mid (1 \;\; 2)\, x > 4.5\}$ and $D_2 = \{x \mid (1 \;\; 2)\, x < 4.5\}$. The system is driven between the points $x(0) = \xi_0 = (-1 \;\; 1)' \in D_2$ and $x(1.8) = \xi_T = (0 \;\; 3.5)' \in D_1$ and, according to the definitions in Section 5.3 and Section 5.4, the transition behavior of the regional dynamics system is determined by the automaton shown in Figure 5.6. Interesting here and, in fact, the main incentive for considering a two-regional framework, again, is the comparison of the actual point-to-point problem with the precedent bimodal problem formulations in Chapter 4, which are, in general, represented by transition automata as the one displayed in Figure 5.4. Comparing both automata, Figure 5.4 and Figure 5.6, the limited dynamics in the preliminary approach of Chapter 4, not allowing a "bounce back" at the boundary, become apparent on this level of abstraction by the omitted self-loops e_{11} and e_{22}.

The regional dynamics determining the hybrid behavior of the continuous state $x(t) = \big(x_1(t) \;\; x_2(t)\big)' \in \mathcal{X}$ are given by

$$\dot{x}(t) = \begin{cases} \begin{pmatrix} 0 & 0.25 \\ -3 & -0.5 \end{pmatrix} x(t) + \begin{pmatrix} -10 \\ 100 \end{pmatrix} u(t), & x(t) \in D_1 \\ \begin{pmatrix} 0.5 & 0.1 \\ -10 & -0.5 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(t), & x(t) \in D_2. \end{cases} \tag{5.32}$$

Moreover, the final time is $T = 1.8$, with the maximum number of switches being given by $N = 3$. The particular cost function (5.19) under consideration is the control energy of the control signal, that is, $\ell(x(t), u(t)) = u(t)^2$, where an unconstrained input $u(t) \in \mathbb{R}$ is assumed.

The numerical solution is obtained by discretizing the time interval $[0, T]$ into 18 equally spaced temporal steps and by discretizing the switching manifold $m_{(1,2)}: \ (1 \ \ 2)\ x = 4.5$ into 40 equally spaced spatial steps over the interval $x_1 \in [-2,\, 2]$. In Figure 5.7, the intermediate results of the Hybrid Bellman calculations (5.25), i.e., the optimal solutions for a given fixed number of switches $M \in \{1, 2, 3\}$, are shown together with their corresponding costs $V^M(\xi_0, q_2, \xi_T, q_1, T)$. The resulting optimal trajectories $(x_1^*(\cdot),\, x_2^*(\cdot))$ are depicted by solid, blue lines; the dashed line represents the boundary $m_{(1,2)}$ and the circles depict the particular switching points. The phenomenon of "bouncing back" at the switching manifold is observed in the cases $M = 2$ and $M = 3$, Figure 5.7(b) and Figure 5.7(c), respectively. Finally, the optimal solution with its associated cost $W^3(\xi_0, q_2, \xi_T, q_1, T)$ is found from the above results by equation (4.9). In this case, the optimal solution is obtained when only one crossing of the switching surface takes place, with the corresponding optimal cost being

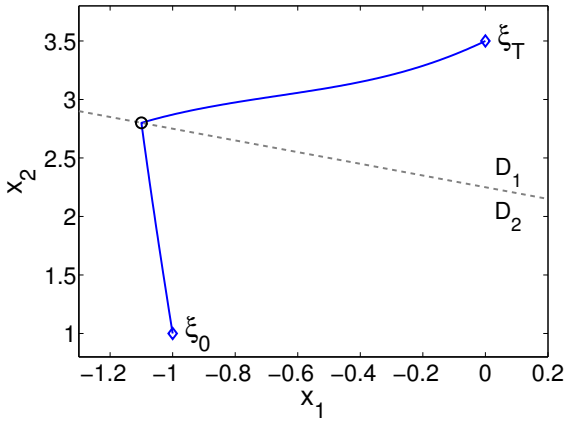
$$W^3(\xi_0, q_2, \xi_T, q_1, T) = V^1(\xi_0, q_2, \xi_T, q_1, T) \approx 3.865\,. \tag{5.33}$$

However, of course, examples can be constructed for which the optimal solution does, in fact, involve multiple switches as, for example, shown in [14]. □

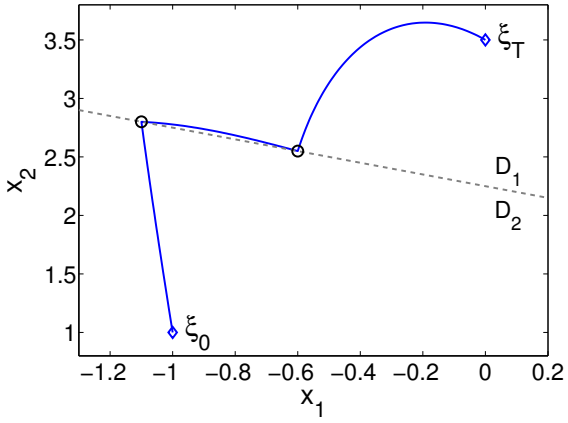
With the goal of finally presenting a *multi*regional situation, indeed, consisting of more than two regions and, especially, illustrating the influence of the global accessibility relations in the proposed solution algorithm, Theorem 5.1, a state space $\mathcal{X}$ partitioned into three regions is considered, see Figure 5.3:

Example 5.2. In this example, the state space $\mathcal{X}$ is divided by two parallel lines,

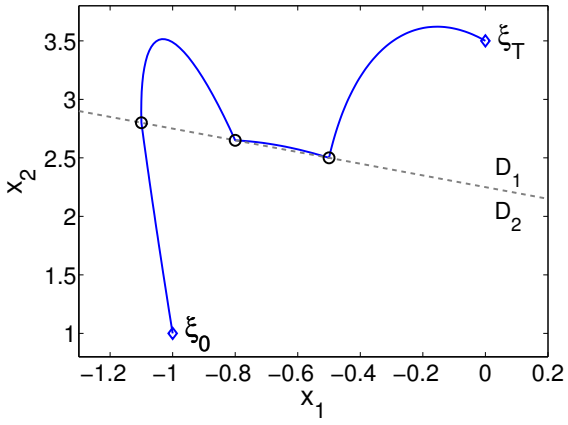
$$m_{(1,2)}: \ (-1 \ 1)\ x = -0.5 \quad \text{and} \quad m_{(2,3)}: \ (-1 \ 1)\ x = -2, \tag{5.34}$$



(a) $M = 1\,,\quad V^1(\xi_0, q_2, \xi_T, q_1, T) \approx 3.865$



(b) $M = 2\,,\quad V^2(\xi_0, q_2, \xi_T, q_1, T) \approx 3.866$



(c) $M = 3\,,\quad V^3(\xi_0, q_2, \xi_T, q_1, T) \approx 3.867$

Figure 5.7: Optimal Trajectories $(x_1^*(\cdot),\, x_2^*(\cdot))$ Corresponding To Example 5.1 – Each For a Given Fixed Number of Switches $M \in \{1, 2, 3\}$

yielding to a three-regional framework composed of the domains

$$D_1 = \{x \mid -0.5 < (-1 \;\; 1)\; x\}\,, \tag{5.35}$$

$$D_2 = \{x \mid \;\; -2 < (-1 \;\; 1)\; x < -0.5\}\,, \quad \text{and} \tag{5.36}$$

$$D_3 = \{x \mid \;\; -2 > (-1 \;\; 1)\; x\}\,. \tag{5.37}$$

The system is driven between the points

$$\xi_0 = (-0.5 \;\; 0)' \in D_1 \qquad \text{and} \qquad \xi_T = (-1.1 \;\; -4)' \in D_3 \tag{5.38}$$

under the regional dynamics

$$\dot{x}(t) = \begin{cases} \begin{pmatrix} -1 & 2 \\ -5 & 1 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(t), & x(t) \in D_1 \\ \begin{pmatrix} 0.5 & 1 \\ -0.05 & -0.25 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(t), & x(t) \in D_2 \\ \begin{pmatrix} 0 & 0.25 \\ -3 & -0.1 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(t), & x(t) \in D_3 \end{cases} \tag{5.39}$$

during a time interval $\mathcal{T} = [0, 6.7]$, where $u(t) \in \mathbb{R}$. As regards the control objective (5.19), the same function as in Example 5.1 is used. Given the upper bound $N = 2$ on the maximum number of switches along a hybrid trajectory, according to the transition automaton in Figure 5.3 representing the proposed geometric structure (5.35)–(5.38), the relevant languages are given by $L_m(0, A) = \emptyset$, $L_m(1, A) = \emptyset$, and $L_m(2, A) = \{e_{12}e_{23}\}$. Consequently, the only feasible discrete control input is $w = e_{12}e_{23}$ resulting in a significantly constrained infimization (5.25).

The numerical solution is obtained by discretizing the time interval $[0, T]$ into 67 equally spaced temporal steps and the switching manifolds $m_{(1,2)}$ and $m_{(2,3)}$ into 40 equally spaced spatial steps over the interval $x_1 \in [-1.5, 2.5]$. The Hybrid Bellman recursion, Theorem 5.1, yield to the optimal hybrid execution depicted in Figure 5.8 (solid, blue line) with the associated cost

$$W^2(\xi_0, q_1, \xi_T, q_3, T) \approx 0.039. \tag{5.40}$$

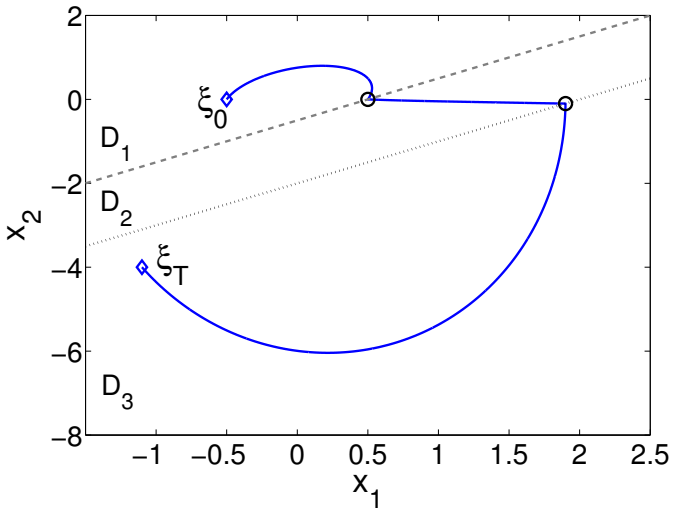
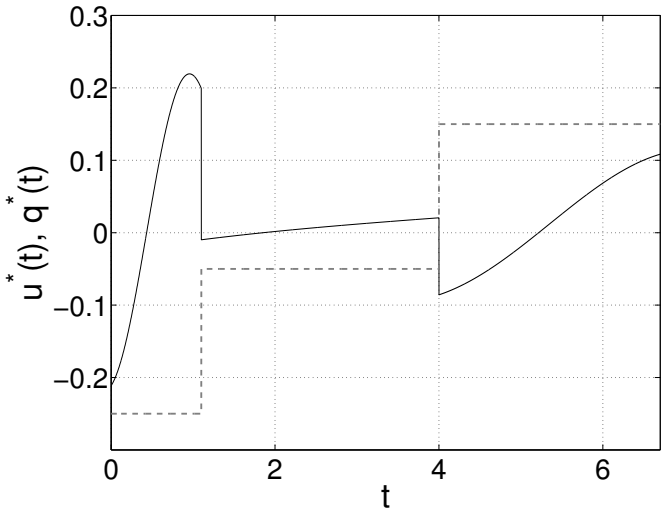
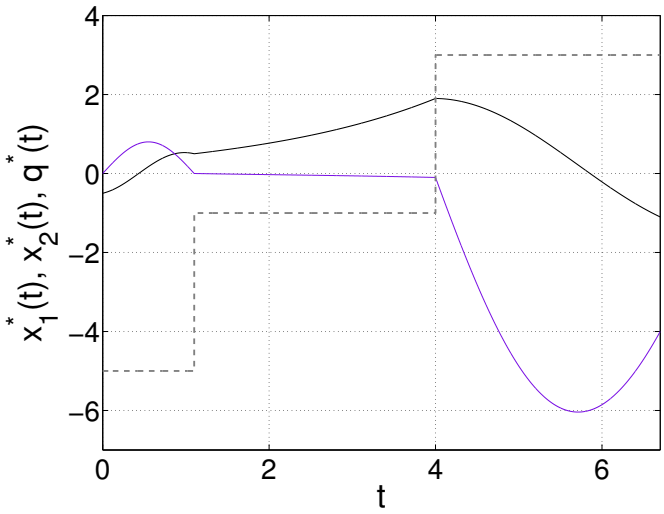


Figure 5.8: Optimal Curve $(x_1^*(\cdot),\, x_2^*(\cdot))$ Corresponding To Example 5.2

The dashed and dotted lines in Figure 5.8 present the switching manifolds $m_{(1,2)}$ and $m_{(2,3)}$, respectively, and the circles display the switching points. In addition, in Figure 5.9, the optimal input $u^*(\cdot)$ and the optimal states $x_1^*(\cdot),\ x_2^*(\cdot)$ are plotted over the time interval $\mathcal{T} = [0,\, 6.7]$. Here, the dashed gray line represents the discrete state $q^*(\cdot)$, where the upper level corresponds to the value q_3, the intermediate stage to q_2, and the lower level to q_1. □



(a) Optimal Input $u^*(\cdot)$



(b) Optimal States $x_1^*(\cdot)$ and $x_2^*(\cdot)$, black line and purple line, respectively

Figure 5.9: Optimal Trajectories Corresponding To Example 5.2

CHAPTER VI

GENERALIZATIONS AND EXTENSIONS

After having shown the core result of this work, the Hybrid Bellman Equation for multimodal systems, which solves, in particular, the point-to-point problem associated with a regional dynamics system, this chapter places special emphasis on the universal validity of the derived automaton-based Dynamic Programming recursion, which applies, in fact, to a much huger class of hybrid optimal control problems characterized by more general dynamic regimes, differently defined geometric structures, and special cost functions.

The previously presented Dynamic Programming approach, cf. Section 5.5 and Section 5.6, allows interesting generalizations and extensions in the formulation of the hybrid optimal control problem. In the following, three different aspects, concerning the expansion of the precedent results, are highlighted: First, in Section 6.1, contrary to the time-invariant approach in Chapter 4 and Chapter 5, dynamic regimes $f_i(x, u, t),\ i \in \mathcal{I}$ explicitly depending on the time t are taken into account and their influence on the recursion equation (5.25) is studied. Central to Section 6.2 are considerations on further possibilities of defining an appropriate control objective, i.e., another cost function to be minimized. Finally, Section 6.3 shows that the assumptions on the geometric structure given in Section 5.2 can be relaxed and even completely different geometries can be approached with a similar Bellman recursion scheme.

Of course, the extensions mentioned in the following sections can be combined at any time. Moreover, the subsequent listed generalizations do not claim to be exhaustive at all; faced with any optimal control problem, it is always recommendable to check, if the idea of Dynamic Programming, illustrated in Section 2.2, leads to a computable, efficient solution algorithm.

6.1 The System Dynamics

This section focuses on the continuous state's behavior within a region $D_i,\ i \in \mathcal{I}$, which is determined by the vector field f_i, and explains an obvious way of extending the previous considerations of Chapter 4 and Chapter 5.

6.1.1 Time-Variant Regional Dynamics

An easy to accomplish generalization in the multimodal framework, introduced in Section 5.2, is the treatment of time-variant dynamics, which yields solely to time-dependent cost functions in the recursion relation (5.25).

At first, it is worth mentioning that, in fact, the bimodal and multimodal (point-to-point) problem formulations, Problem 4.1 and Problem 5.1, respectively, already model the dynamics associated with each individual region $D_i,\ i \in \mathcal{I}$ in a very general way. Compared to some former approaches solving these and other hybrid optimal control problems only in the case of linear dynamical behavior, see e.g. [10], in Chapter 4 and Chapter 5 nonlinear dynamic regimes $\dot{x}(t) = f_i(x(t), u(t))$ are assumed. However, the Dynamic Programming approach solving the multimodal point-to-point problem even applies for the more general time-variant nonlinear regional dynamics

$$\dot{x}(t) = f_i(x(t), u(t), t) \quad \text{if} \ \ x(t) \in D_i,\ i \in \mathcal{I}. \tag{6.1}$$

In this case, the previously defined costs $c(\xi_1,\, q_{i_1},\, \xi_2,\, q_{i_2},\, \Delta)$, cf. Definition 4.1, and $V^M(\xi_1,\, q_{i_1},\, \xi_2,\, q_{i_2},\, \tau)$, cf. Definition 4.2, representing important concepts in the derivation of the Hybrid Bellman Equation (5.25), are also time-dependent; that is, they depend not only on the traveling interval Δ or τ, but on the exact start time t_1, where $x(t_1) = \xi_1$, and the final time, where $x(t_2) = \xi_2$. Consequently, the function $c(\xi_1,\, q_{i_1},\, t_1,\, \xi_2,\, q_{i_2},\, t_2)$ is introduced as the infimum of the costs associated with driving the system from $x(t_1) = \xi_1 \in D_{i_1} \cup \partial D_{i_1}$ to $x(t_2) = \xi_2 \in D_{i_1} \cup \partial D_{i_1},\ i_1 \in \mathcal{I}$ without leaving $D_{i_1} \cup \partial D_{i_1}$ and without a switching taking place. And analogously, $V^M(\xi_1,\, q_{i_1},\, t_1,\, \xi_2,\, q_{i_2},\, t_2)$ denotes the infimum of the costs of going from $x(t_1) = \xi_1 \in \mathcal{X}$ to $x_2(t_2) = \xi_2 \in \mathcal{X}$ using exactly M switches and starting in

region D_{i_1}. With these "extended" definitions, the corresponding Hybrid Bellman Equation solving the point-to-point problem in the time-variant regional dynamics framework (6.1) is immediately given by (5.25)

$$V^K(\xi_1,\, q_{i_1},\, t_1,\, \xi_2,\, q_{i_2},\, t_2)$$

$$= \inf_{t\in(t_1,t_2)} \inf_{\xi\in m_{(i_1,j)}} \inf_{j\in\mathcal{I}} \left\{ c(\xi_1, q_{i_1}, t_1, \xi, q_j, t) + V^{K-1}(\xi, q_j, t, \xi_2, q_{i_2}, t_2) \right\} \tag{6.2}$$

with the additional global language constraints (5.26) and (5.27), cf. Theorem 5.1. Moreover, because of the system's time variance, the initial and final condition of the corresponding point-to-point problem, cf. (5.23), are in general $x(t_0) = \xi_0$ and $x(t_f) = \xi_f$, respectively. The above straight forward conclusion results in a huger computational effort while preparing the costs $c(\xi_1,\, q_{i_1},\, t_1,\, \xi_2,\, q_{i_2},\, t_2)$, which have to be calculated for all $(t_1, t_2) \in \mathcal{T} \times \mathcal{T}$ with $\mathcal{T} = [t_0, t_f]$; nevertheless, the effort associated with the recursion (6.2) itself is the same as in the time-invariant case.

The time variance can also be introduced in the cost function (5.19) or, more precisely, in $\ell(x(t), u(t))$. This and other variations concerning the control objective of the hybrid point-to-point problem are mentioned in the subsequent section.

6.2 The Cost Function

The actual hybrid optimal solution $x^*(\cdot)$ is a direct consequence of the way the control objective of the hybrid optimal control problem is defined; in the following, four different modifications of the original cost function (5.19) are briefly presented.

6.2.1 Transition Costs

An additional feature, whose influence on the hybrid optimal solution represents certainly an interesting issue to investigate, is the introduction of transition costs $c_e(x,t),\ e \in \mathcal{E}$ assigning a weight to each switching point $\left(t_s^k, \xi_s^k\right)$ along a hybrid

trajectory. As a consequence, the control objective in the multimodal point-to-point problem, Problem 5.1, is read as

$$\mathcal{P}_N : \inf_{\substack{u(\cdot)\in\mathcal{U},\\ S(\overline{\tau},w)}} \left\{ \int_0^T \ell(x(t),u(t))\,dt + \sum_{k=0}^{M} c_{e_{i_k j_k}}\left(\xi_s^k, t_s^k\right) \right\} \tag{6.3}$$

and the Hybrid Bellman Equation (5.25) shows the following slight change:

$$V^K(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau) = \inf_{t\in(0,\tau)} \inf_{\xi\in m_{(i_1,j)}}$$

$$\left\{ \inf_{j\in\mathcal{I}} \left\{ c(\xi_1, q_{i_1}, \xi, q_j, t) \;+\; c_{e_{i_1 j}}(\xi, t) \;+\; V^{K-1}(\xi, q_j, \xi_2, q_{i_2}, \tau - t) \right\} \right\}, \tag{6.4}$$

where again the global accessibility relations, (5.26) and (5.27), constrain the infimization.

Remarkably, this notable conceptual change, allowing to give preference to certain transitions and, in fact, gain influence on the number of switches along the hybrid optimal trajectory, does not change the algorithmic and computational procedure by any means.

6.2.2 Time Optimality

A standard optimal control objective is the time-optimal performance of a system's dynamic behavior. In the context of the considered multimodal point-to-point problem, cf. Section 3.1, that means, finding *the fastest way* of going from a given initial state ξ_0 to a fixed final state ξ_f or, more precisely,

$$\mathcal{P}_N : \inf \inf_{\substack{u(\cdot)\in\mathcal{U},\\ S(\overline{\tau},w)}} \int_0^T 1\;dt \;=\; \inf_{\substack{u(\cdot)\in\mathcal{U},\\ S(\overline{\tau},w)}} T \tag{6.5}$$

subjected to equations (5.20)–(5.23) as given in Problem 5.1.

Important to emphasize is that, in contrast to the precedent bimodal and multimodal problem formulations, Problem 4.1 and Problem 5.1, respectively, the time horizon $T \in [0, \infty)$ is not specified in advance, but is, in fact, the result, i.e. the value function, of the hybrid optimal control problem (6.5). Nevertheless, the previous hierarchic approach provides an ideal basis for dealing with this time-optimal point-to-point problem. An appropriate definition of the cost $c(\cdot, \cdot, \cdot, \cdot, \cdot)$

and the cost-to-go function $V^{\cdot}(\cdot, \cdot, \cdot, \cdot, \cdot)$, cf. Section 4.4, which are, indeed, time values in this special case, set the stage for the time-optimal Hybrid Bellman Equation. Let $c(\xi_1, q_{i_1}, \xi_2, q_{i_2})$ denote the infimum of the traveling times needed to drive the system's continuous state from $\xi_1 \in D_{i_1} \cup \partial D_{i_1}$ to $\xi_2 \in D_{i_1} \cup \partial D_{i_1}$, $i_1 \in \mathcal{I}$ without leaving $D_{i_1} \cup \partial D_{i_1}$ and without a switching taking place and $V^M(\xi_1, q_{i_1}, \xi_2, q_{i_2})$ the infimum of the traveling times needed to drive the system's continuous state from $\xi_1 \in \mathcal{X}$ to $\xi_2 \in \mathcal{X}$ using exactly M switches and starting in region D_{i_1}. Then, the time-optimal recursion relation follows immediately from the Hybrid Bellman Equation in Section 5.6:

$$V^K(\xi_1, q_{i_1}, \xi_2, q_{i_2})$$

$$= \inf_{\xi \in m_{(i_1,j)}} \inf_{j \in \mathcal{I}} \left\{ c(\xi_1, q_{i_1}, \xi, q_j) + V^{K-1}(\xi, q_j, \xi_2, q_{i_2}) \right\} \quad (6.6)$$

with the global constraints given by equations (5.26) and (5.27). In this case, there is no need for an infimization over the time $t \in (0, \tau)$ as previously observed in the original bimodal and multimodal results, Theorem 4.1 and Theorem 5.1, respectively. However, apart from that, the parallel structure of both approaches is evident.

6.2.3 Time-Variant Cost Function

The introduction of a time-variant cost function in the multimodal point-to-point problem, Problem 5.1, yielding to the control objective

$$\mathcal{P}_N : \inf_{\substack{u(\cdot) \in \mathcal{U}, \\ S(\overline{\tau}, w)}} \int_{t_0}^{t_f} \ell(x(t), u(t), t)\, dt, \quad (6.7)$$

where the time interval $\mathcal{T} = [t_0, t_f]$ under consideration is specified *a priori*, results in exactly the same generalized recurrence relation (6.2) as the considerations on time-variant dynamics in Section 6.1.1. Because of the problems' inherent time variance, in both cases, the cost functions $c(\xi_1, q_{i_1}, t_1, \xi_2, q_{i_2}, t_2)$ and $V^M(\xi_1, q_{i_1}, t_1, \xi_2, q_{i_2}, t_2)$, which are fundamental to the Dynamic Programming approach, depend on the exact start and final time, t_1 and t_2, respectively. Besides, all further conclusions stated in Section 6.1.1 apply in an identical way also

to the above generalized assumption (6.7).

6.2.4 Individual Regional Costs

Besides Section 6.2.1, another exciting extension of the original approach, which provides an additional opportunity of characterizing the hybrid point-to-point problem and does not have any effect on the computational effort associated with accomplishing the recursive algorithm, represents the assumption of individual cost functions $\ell_i(x(t), u(t)),\ i \in \mathcal{I}$ for each domain D_i. As a result, the new performance goal in the regional dynamics optimal control problem, cf. Problem 5.1, is

$$\mathcal{P}_N : \inf_{\substack{u(\cdot)\in\mathcal{U},\\ S(\overline{\tau},w)}} \left\{ \sum_{k=0}^{M} \int_{t_s^{k+1}}^{t_s^k} \ell_{i_{k+1}}(x(t), u(t))\ dt \right\}, \tag{6.8}$$

while satisfying the constraints (5.20)–(5.23). In fact, recalling the hierarchic structure of the multimodal point-to-point problem introduced in Section 5.1, this new objective (6.8) influences only the lowest level of control, where in each case the appropriate function $\ell_{i_1}(x(t), u(t))$ must be chosen in the calculation of the costs $c(\xi_1,\, q_{i_1},\, \xi_2,\, q_{i_2},\, \Delta)$, cf. Definition 4.1. he higher levels of abstraction, however, remain completely unaffected.

This slight change in the definition of the control objective as given by (6.8) leads to remarkable consequences regarding the definition of the system's regional dynamics or, more precisely, the dimension of the input $u(\cdot)$ and the associated set of admissible values $u(t) \in,\ t \in [0, T]$ as specified in Assumption 4.1. Already in the original multimodal problem statement, Problem 5.1, different constraint sets for each domain $D_i,\ i \in \mathcal{I}$ can be considered by replacing the input constraint in (5.19) with

$$u(t) \in U_{i_{k+1}}, \qquad t \in \left[t_s^k, t_s^{k+1}\right]. \tag{6.9}$$

However, under the assumption of individual cost functions $\ell_i(x(t), u(t)),\ i \in \mathcal{I}$ in each contributing domain D_i, it is even possible to vary the dimension of the

input $u(t) \in U_i \subseteq \mathbb{R}^{n_i},\ n_i \in \mathbb{N}$ depending on the region D_i the continuous state $x(t)$ is evolving in.

6.3 The Geometric Structure

The state space $\mathcal{X}$ and its partitioning into various domains D_i, which are associated with different dynamical systems $\dot{x} = f_i(x, u)$, together with the given starting point ξ_0 and the final state ξ_T set the geometric framework of the considered multimodal point-to-point problem, Problem 5.1. A concise and neat model of this complex structure is given by the transition automaton introduced in Section 5.4 highlighting, in particular, the connections between the individual regions.

Subsequently, based on the automaton representation, different ways of relaxing the above listed characteristics are indicated. Especially, the main assumption and restriction in the definition of a regional dynamics system, the direct connection between the dynamic behavior $\dot{x} = f_i(x, u)$ and the corresponding region D_i, cf. (5.5), that is, the dependence of the discrete state q on the continuous state x is abolished in the considerations of Section 6.3.3 and Section 6.3.4. Indeed, the following sections only present a rough sketch of the ideas using the previous hierarchic solution scheme to solve the generalized problem statements. Future work will include more detailed definitions, an appropriate literature overview, examples for applications, and numerical investigations and results.

6.3.1 Free Final State

As indicated in Figure 6.1, this section considers a regional dynamics optimal control problem without terminal constraints; that means, the end point $x(T) = \xi_T \in D_i,\ i \in \mathcal{I}$ is not specified in advance, but represents, in fact, an additional optimization parameter. As a consequence, every region $D_i,\ i \in \mathcal{I}$ must be regarded as a possible final domain $D_{i_T} :\ x(T) \in D_{i_T}$ and an appropriate transition automaton A, cf. equation (5.11), redefines the set of marked states $\mathcal{D}_m$, previously given by $\mathcal{D}_m = \{i_T \mid \xi_T \in D_{i_T}\}$, as $\mathcal{D}_m = \mathcal{I}$. Figure 6.1 shows the automaton corresponding to the four-regional example in Figure 5.5 for an unconstrained final

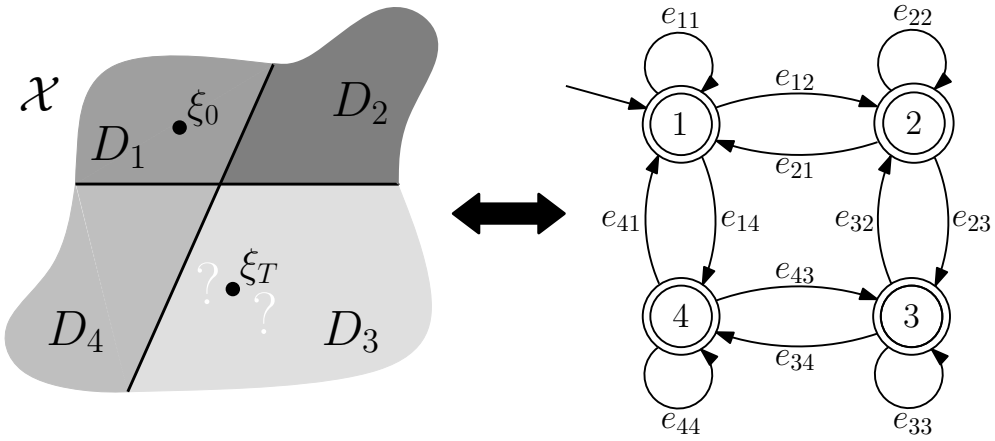


Figure 6.1: Transition Automaton Corresponding To A Regional Dynamics Optimal Control Problem With Free Final State

state ξ_T.

Again, the hierarchical structure proposed in Chapter 5 serves as a valuable basis for a fundamental understanding and successful solution of the considered hybrid optimal control problem, where, once more, on the highest level of control, the global accessibility relations, provided by the automaton's marked language, specify possible and allowed, see Remark 5.2, transition sequences between the individual regions. This global knowledge is still an essential feature in the solution procedure, even though the final state is free, what leads solely to a slightly changed meaning of the underlying languages. In fact, in this case, the marked language is simply given by $L_m(A) = L(A)$. Finally, the Dynamic Programming approach to the free-final-state problem can be formulated in terms of the previous multimodal point-to-point solution presented in Theorem 5.1. In accordance with the new problem statement, the notation $V^M(\xi_1, q_{i_1}, \tau)$ is used to denote the appropriate cost-to-go function, which is defined in analogy to Definition 4.2 as the infimum of the costs of originating at the point $\xi_1 \in \mathcal{X}$ and evolving in a given regional dynamics system during a time interval $[0, \tau]$ using exactly M switches and starting out in region D_{i_1}. Using this definition, the modified Hybrid Bellman

Equation is given by

$$V^K(\xi_1, q_{i_1}, \tau)$$

$$= \inf_{t\in(0,\tau)} \inf_{\xi\in m_{(i_1,j)}} \inf_{j\in\mathcal{I}} \Big\{ c(\xi_1, q_{i_1}, \xi, q_j, t) + V^{K-1}(\xi, q_j, \tau - t) \Big\} \quad (6.10)$$

where the global constraints to be satisfied, (5.26) and (5.27), take the above mentioned changes in the definition of the automaton into account. Furthermore, the recursion's initial condition, cf. (5.28), involves additional minimizations

$$V^0(\xi_1, q_{i_1}, \tau) = \min_{i_2\in\mathcal{I}} \inf_{\xi_2\in D_{i_2}} \Big\{ c(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau) \Big\}. \quad (6.11)$$

Indeed, from a computational point of view, only the first step in the recursive backwards procedure specified by (6.11) differs from the multimodal algorithm.

6.3.2 Segmentation of Switching Manifolds

An exciting generalization, especially with regard to the multi-agent example in Chapter 7, is the interpretation of a switching manifold $m_{(i,j)},\ i,j \in \mathcal{I}$ as a union of constituent subcomponents $\tilde{m}^k_{(i,j)}$:

$$m_{(i,j)} = \bigcup_{k=1}^{\beta_{ij}} \tilde{m}^k_{(i,j)}, \quad i,j \in \mathcal{I}. \quad (6.12)$$

Treating each subcomponent $\tilde{m}^k_{(i,j)},\ k \in \{1, \ldots, \beta_{ij}\}$ individually in the hybrid optimal control approach to the multimodal point-to-point problem, that is, introducing, in particular, separate transition labels for every contributing component $\tilde{m}^k_{(i,j)}$, provides the opportunity to force the continuous state to pass through a certain boundary segment and not to cross other parts of the switching manifold. In fact, in view of a multi-agent problem, these considerations allow to characterize sections along a boundary, which the agents are unable or not permitted to pass. Of course, the multi-agent example is only one example out of a huge number of interesting applications.

In order to define this idea more precisely, the following changes are proposed in the definition of the transition automaton (5.11): In one-to-one correspondence

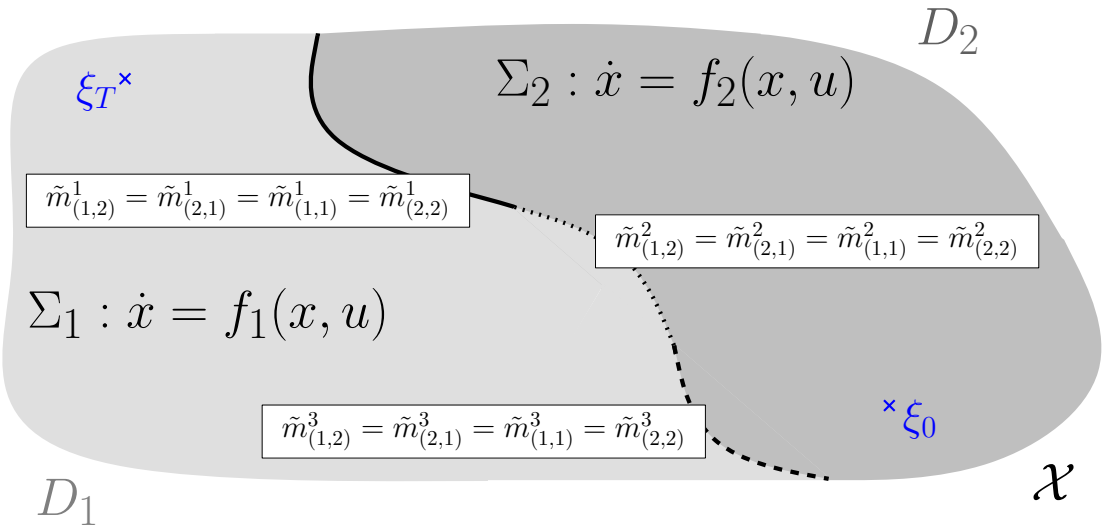


Figure 6.2: Bimodal Example with Segmented Switching Manifolds

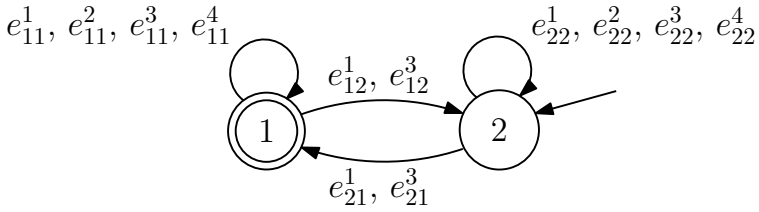


Figure 6.3: Transition Automaton Corresponding to the Geometric Framework in Figure 6.2

with the set of switching manifold subcomponents

$$\mathcal{M} = \left\{ \tilde{m}^{\alpha_{ij}}_{(i,j)} \;\middle|\; i, j \in \mathcal{I}, \;\; \alpha_{ij} \in \{1, \ldots, \beta_{ij}\} \right\}, \tag{6.13}$$

the set of events $\mathcal{E}$ is extended to

$$\mathcal{E} = \left\{ e^{\alpha_{ij}}_{(i,j)} \;\middle|\; i, j \in \mathcal{I}, \;\; \alpha_{ij} \in \{1, \ldots, \beta_{ij}\} \right\}. \tag{6.14}$$

Moreover, on this larger set of events $\mathcal{E}$, the transition function g is defined as

$$g(i, e^{\alpha_{ij}}_{ij}) = j \qquad \text{if the event } e^{\alpha_{ij}}_{(i,j)} \in \mathcal{E} \text{ is allowed and } \tilde{m}^{\alpha_{ij}}_{(i,j)} \neq \emptyset. \tag{6.15}$$

In all other cases, the function g is not defined.

Figure 6.2 shows a bimodal example, where the switching manifold $m_{(1,2)}$ is divided into three subcomponents, $\tilde{m}^1_{(1,2)}$, $\tilde{m}^2_{(1,2)}$, and $\tilde{m}^3_{(1,2)}$. Moreover, for the sake of

simplicity and clearness, the illustration assumes $\tilde{m}^k_{(1,2)} = \tilde{m}^k_{(2,1)} = \tilde{m}^k_{(1,1)} = \tilde{m}^k_{(2,2)}$ for $k = 1, 2, 3$. The part $\tilde{m}^4_{(i,i)},\ i = 1, 2$ is given by $\tilde{m}^4_{(i,i)} = m_{(i,i)} \setminus \left\{ \bigcup_{k=1}^{3} \tilde{m}^k_{(i,i)} \right\}$. The suggested automaton, illustrated in Figure 6.3, controls the continuous state's execution in the way that a transition between the regions is only allowed on the segments $\tilde{m}^1_{(1,2)} = \tilde{m}^1_{(2,1)}$ and $\tilde{m}^3_{(1,2)} = \tilde{m}^3_{(2,1)}$. In Chapter 7, where a a multi-agent problem is considered, part $\tilde{m}^2_{(1,2)}$ may represent a wall or another object, which makes a crossing impossible for the considered group of agents. Note that the "bounce back" behavior, depicted by self-loops in the automaton representation, Figure 6.3, is possible at every point on $m_{(1,2)}$.

In fact, the presented extension to the original multimodal problem definition, Problem 5.1, provides an additional feature supporting a more appropriate and elaborated way of modeling real system as previously indicated by the consideration of Figure 6.2 and Figure 6.3 as a geometric framework for a multi-agent application. Moreover, this powerful modeling tool only causes very small modifications in the formulation of the multimodal point-to-point problem (MPTPP), Problem 5.1, and in the recursive solution procedure, Theorem 5.1. In the following, based on the definitions (6.12)–(6.15), both, Problem 5.1 and Theorem 5.1, are restated for the case of segmented boundaries:

Problem 6.1 (The MPTPP With Segmented Switching Manifolds).

$$\mathcal{P}_N : \inf_{\substack{u(\cdot)\in\mathcal{U},\\ S(\overline{\tau},w)}} \int_0^T \ell(x(t), u(t))\, dt \tag{6.16}$$

subject to, for $0 \leq M \leq N$,

- *the geometric structure*

$$w = e^{\alpha_1}_{i_1 j_1}\, e^{\alpha_2}_{i_2 j_2}\, \ldots e^{\alpha_M}_{i_M j_M} \in L_m(M, A), \tag{6.17}$$

- *the discrete dynamics*

$$q_{j_k} = \Gamma\, \left(q_{i_k}, e^{\alpha_k}_{i_k j_k}\right) \tag{6.18}$$

at a switching time t^k_s, *where* $1 \leq k \leq M$, *yielding to the discrete-state dynamics* $q(t) = q_{i_k},\ t \in \left[t^{k-1}_s,\, t^k_s\right)$ *with* $0 < k \leq M+1$ *and* $i_{M+1} = j_M$,

- *the continuous-state dynamics*

$$\dot{x}_{i_k}(t) = f_{i_k}(x_{i_k}(t), u(t)), \quad t \in \left[t_s^{k-1},\, t_s^k\right), \tag{6.19}$$

where $0 < k \leq M+1$ *and* $i_{M+1} = j_M$,

- *and the corresponding initial and final conditions*

$$\begin{aligned} x(0) &= x_{i_1}(t_s^0) &&= \xi_0 \quad \in D_{i_0}, \\ x_{i_{k+1}}(t_s^k) &= \lim_{t \to t_s^k} x_{i_k}(t) &&= \xi_s^k \quad \in \tilde{m}^{\alpha_k}_{(i_k, j_k)}, \\ x(T) &= x_{i_{M+1}}(t_s^{M+1}) &&= \xi_T \quad \in D_{i_T}, \end{aligned} \tag{6.20}$$

if $0 < k \leq M$.

□

With a generalized version of the function *end* : $\mathcal{E}^* \setminus \{\epsilon\} \rightarrow \mathcal{I}$, cf. equation (5.24), defined on the larger set of events $\mathcal{E}$, given by (6.14), as

$$end(s) = end(e^{\alpha_1}_{i_1 j_1}\, e^{\alpha_2}_{i_2 j_2}\, \ldots e^{\alpha_k}_{i_k j_k}) = j_k, \tag{6.21}$$

the Hybrid Bellman Equation for multimodal systems with segmented switching manifolds follows immediately.

Theorem 6.1 (The Hybrid Bellman Approach To Problem 6.1). *Assume that all hypotheses for the existence and uniqueness of regional dynamics hybrid systems hold and that all infima exist in the definition of the hybrid value functions* $V^{\cdot}(\cdot,\cdot,\cdot,\cdot,\cdot)$, *for all admissible argument values, whenever the expressions are finite. Then, with the transition automaton A, given by equation* (5.11) *with the specifications* (6.14) *and* (6.15), *the recurrence relation corresponding to the multimodal point-to-point problem with segmented switching manifolds, Problem 6.1, is expressed by*

$$\begin{aligned} &V^K(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau) \\ &\quad = \inf_{t \in (0,\tau)} \; \inf_{\xi \in m^{\alpha}_{(i_1, j)}} \; \inf_{j \in \mathcal{I}} \; \left\{ c(\xi_1, q_{i_1}, \xi, q_j, t) + V^{K-1}(\xi, q_j, \xi_2, q_{i_2}, \tau - t) \right\} \end{aligned} \tag{6.22}$$

such that

$$e \;=\; e^{\alpha}_{i_1 j}\,, \qquad w \;\in\; F_{K-1}(M,A)\,, \tag{6.23}$$

$$\mathit{end}(w) \;=\; i_2\,, \qquad ew \;\in\; F_K(M,A)\,. \tag{6.24}$$

This relation holds for $0 < K \leq M$. *The initial condition of the recursive scheme is given by*

$$V^0(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau) \;=\; c(\xi_1, q_{i_1}, \xi_2, q_{i_2}, \tau). \tag{6.25}$$

□

The computational procedure solving the modified point-to-point problem defined by Problem 6.1 follows exactly the same steps as the original multimodal approach, see Section 5.7 for a detailed description. The segmentation may solely increase the number of words contained in the language sets $L_m(M,A)$ and $F_K(M,A)$. However, generally speaking, compared to the previously considered multimodal approach of Chapter 5, the effort needed to accomplish the recursive algorithm (6.22) does not change significantly. In particular, it is important to highlight that a segmentation of the boundaries does not affect the size of discrete set of points resulting from the computationally necessary boundary discretization.

6.3.3 State-Independent Discrete Modes

This section goes beyond the actual subject of the presented work, indicated by the thesis' title "Optimal Control of Hybrid Systems with Regional Dynamics", in that the assumption of regional dynamics, that is, the strong connection between the dynamic behavior $\dot{x} = f_i(x,u)$ and the corresponding region D_i, cf. equation (5.5), is abolished. However, a brief investigation of this generalized setup highlights, once again, the general validity of the proposed hierarchic Dynamic Programming approach.

First and foremost, the question, which dynamic regime $f_k,\ k \in \mathcal{I}$ defines the continuous state's execution at a time instant t, is no longer answered by the region D_k the continuous state $x(t)$ is currently evolving in, but, in fact, depends solely on the discrete state value $q(t) = q_k$; i.e., the notation $x_k(t)$, introduced in Section 4.2, is used in the following way

$$\dot{x}_k(t) = f_k(x_k(t), u(t)) \quad \text{if} \quad q(t) = q_k,\, k \in \mathcal{I}, \tag{6.26}$$

where $\mathcal{I}$ represent a given set of discrete modes. Obviously, the main difference to the previous considerations in Chapter 4 and Chapter 5 is the spatial independence of the discrete state $q:\ q \neq q(x)$. Instead of defining regions and switching boundaries, the (not necessarily connected) sets $\mathcal{M}_{ij},\ i,j \in \mathcal{I}$ are introduced containing the points of the state space $\mathcal{X}$ which allow a transition from mode i to mode j. As illustrated in Figure 6.4, given a continuous starting state ξ_0 and a discrete initial mode i_0, the continuous state's evolution starts at ξ_0 under the dynamics f_{i_0} and a mode transition to another dynamic regime $f_j,\ j \in \mathcal{I}$ may happen, if the state enters a set $\mathcal{M}_{i_0 j},\ j \in \mathcal{I}$. After a fixed time T, the trajectory is forced to reach the *a priori* specified final point ξ_T in mode i_T. Even though the transition behavior of the continuous state is not restricted by an underlying regional framework, a transition automaton A is introduced as a discrete representation of the possible mode transitions e_{ij}. A transition e_{ij} is defined if $\mathcal{M}_{ij} \neq \emptyset,\ i,j \in \mathcal{I}$. Moreover, on this highest level of abstraction additional, restrictive switching rules, cf. Remark 5.2, are taken into account. Figure 6.5 shows one potential, highly limiting automaton representation corresponding to the exemplary illustration in Figure 6.4.

In fact, some little changes in the definitions of Chapter 5 suffice to apply the hierarchic hybrid Bellman approach, Theorem 5.1, equally to the point-to-point problem associated with the above introduced hybrid system with state-independent discrete modes:

- The fundamental concepts, the hybrid Dynamic Programming algorithm, Theorem 5.1, is based on, the cost functions $c(\cdot,\cdot,\cdot,\cdot,\cdot)$ and $V^{\cdot}(\cdot,\cdot,\cdot,\cdot,\cdot)$,

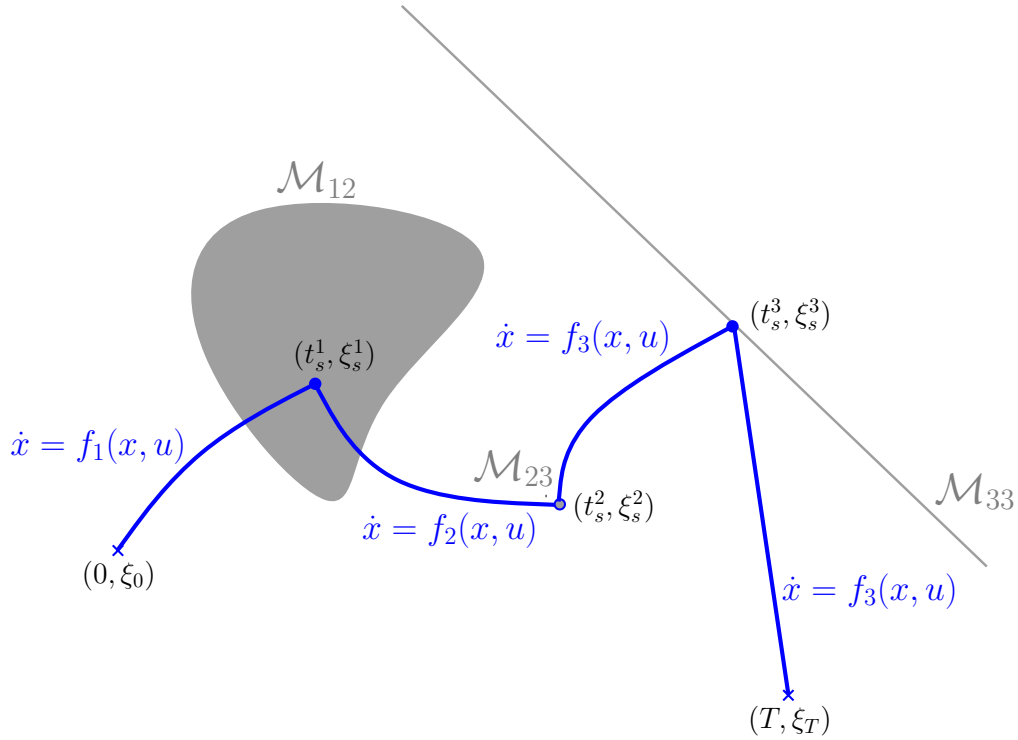


Figure 6.4: Dynamics of a Hybrid System with State-Independent Discrete Modes, given the initial mode $i_0 = 1$ and final mode $i_T = 3$

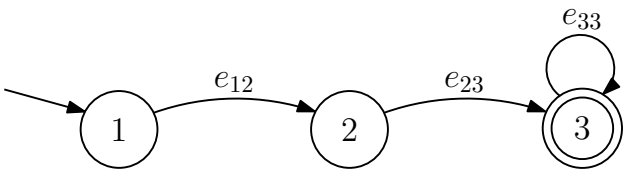


Figure 6.5: Proposed Transition Automaton in the Example of Figure 6.2

are redefined in a region-independent way. Here, the function $c(\xi_1,\, q_{i_1},\, \xi_2,\, q_{i_2}$ denotes the infimum of the costs associated with driving the system from $\xi_1 \in \mathcal{X}$ under the dynamics $\dot{x} = f_i(x,u)$ to $\xi_2 \in \mathcal{X}$ over a time horizon Δ without a switching taking place. The cost-to-go function $V^M(\xi_1,\, q_{i_1},\, \xi_2,\, q_{i_2},\, \tau)$ provides the infimum of the costs of going from $\xi_1 \in \mathcal{X}$ to $\xi_2 \in \mathcal{X}$ during the time horizon τ using exactly M switches and starting in mode i_1 with dynamics $\dot{x} = f_i(x,u)$.

- The given initial and final point, ξ_0 and ξ_T, respectively, are not restricted to lie in a certain region as assumed in Problem 5.1 equation (5.23).

- In the automaton specifications (5.11), in the definition of the (controlled) discrete dynamics, (5.8) and (5.9), and also in Problem 5.1 and Theorem 5.1, the notation $\mathcal{M}_{ij}$ replaces $m_{(i,j)}$.

Hence, the approach of Chapter 5 represents a valuable solution scheme even for a completely different hybrid optimal control problem, which assumes the state independence of the discrete modes $i \in \mathcal{I}$. Finally, interesting to highlight is that the above introduced hybrid system includes the case of a fully discretized state space $\mathcal{X}$, in which each discretization point can be assigned to several (or even to all) sets $\mathcal{M}_{ij},\ i,j \in \mathcal{I}$. This hybrid framework represents a frequently considered hybrid system model, see for example [63].

6.3.4 Controlled Switches

Continuing the previous considerations on hybrid systems with state-independent discrete modes, this section proceeds on the assumption that the discrete control sequence $S\,(\overline{\tau}, w)$, cf. (5.10), with

$$w = e_{i_1 j_1} e_{i_2 j_2} \ldots e_{i_M j_M} \qquad \text{and} \quad \overline{\tau} = (t_s^1,\, t_s^2,\, \ldots,\, t_s^M) \tag{6.27}$$

is given *in advance* and, therefore, no longer an optimal control parameter in the hybrid optimal control problem specified in Section 6.3.3. Consequently, an appropriate Hybrid Bellman Equation, cf. (5.25), which is based on the specifications

of the precedent section, requires neither an infimization over the time $t \in (0, \tau)$, nor an infimization over the set of discrete modes $j \in \mathcal{I}$. With the redefined costs $c(\cdot, \cdot, \cdot, \cdot, \cdot)$ and $V^{\cdot}(\cdot, \cdot, \cdot, \cdot, \cdot)$, given in Section 6.3.3, the recursive relation is simply stated as

$$\begin{aligned} & V^K(\xi_1, q_{i_k}, \xi_T, q_{i_T}, \tau) \\ & = \inf_{\xi \in m_{(i_k, j_k)}} \left\{ c\Big(\xi_1, q_{i_k}, \xi, q_{j_k}, \tau - \big(T - t_s^k\big)\Big) + V^{K-1}\Big(\xi, q_{j_k}, \xi_T, q_{i_T}, T - t_s^k\Big) \right\}, \end{aligned} \tag{6.28}$$

where $\tau > t_s^k$ and $0 < K \leq M$. The final condition, $x(T) = \xi_T$ with $q(T) = i_T$, as well as the given sequence of controlled switches with (6.27), isdirectly incorporated in the Bellman equation (6.28), where the recursion starts with

$$V^0(\xi_1, q_{i_T}, \xi_T, q_{i_T}, T - t_s^M) \; = \; c(\xi_1, q_{i_T}, \xi_T, q_{i_T}, T - t_s^M). \tag{6.29}$$

6.4 Conclusions

In brief, the hierarchic structure introduced in Chapter 5 to solve the multimodal point-to-point problem establishes, in fact, an excellent basis for a successful approach to a huge variety of modified and generalized hybrid optimal control problems.

The three levels of abstraction, namely, the transition automaton offering global accessibility relations on the highest level of control, the standard (non-hybrid) state-constrained optimization problems $c(\cdot, \cdot, \cdot, \cdot, \cdot)$ solved on the lowest level of control, and the recursive Dynamic Programming algorithm bringing both results together and, finally, providing the hybrid optimal trajectory, lay the foundation for all the different problem formulations in the previous sections. However, what has to be adapted in some of the investigated generalizations, is the definition of the conceptually important cost functions, $c(\cdot, \cdot, \cdot, \cdot, \cdot)$ and $V^{\cdot}(\cdot, \cdot, \cdot, \cdot, \cdot)$, as observed in Section 6.1.1, where time-variant system dynamics are introduced, in Section 6.2.2, which deals with the time-optimal control objective, in Section 6.2.3, which considers the time variance in the cost function, and in Section 6.3.3 and

Section 6.3.4, where regional-independent discrete states are taken into account. Moreover, other cases only require a redefinition of the transition automaton; see for example the approach in Section 6.3.1 to the free-final-state problem and the considerations on segmented boundaries in Section 6.3.2. Furthermore, the precise problem statements in Section 6.2.2 and Section 6.3.4 even allow a reduced number of infimizations in the corresponding modified Hybrid Bellman Equation.

Finally, recalling the results of the previous sections, the most significant observation to be made is the general applicability of the proposed hierarchical Dynamic Programming recursion, which, in fact, provides a characterization of global optimality for a significantly larger class of hybrid optimal problems than the originally considered multiregional point-to-point problems.

CHAPTER VII

LEADER-BASED MULTI-AGENT COORDINATION

The powerful operation of the proposed hierarchic Dynamic Programming approach, established in Chapter 4 and Chapter 5, is proven by solving a hybrid multi-agent problem, where the agents' formation and, therefore, the overall system dynamics switches depending on predefined regional decision criteria.

After the doubtlessly very abstract and theoretical considerations in the previous chapters, Chapter 3, 4, and 5, and an even further generalization of the approach in Chapter 6, which represents, of course, a remarkably advantageous characteristic of the derived solution scheme, this chapter aims at actually applying the presented algorithms to a real-world example. Considered in the following is a heterogenous multi-agent system represented by a collection of leaders which dictate the motion of the followers. The hybridity in this example results from a regional-dependent change in the interconnections between the individual agents and their assigned roles as leaders and followers. Focusing on both, the numerical practicability of the recursive algorithm (5.25) and the behavioral characteristics of different multi-agent formations subjected to an optimal control policy, numerous demonstrative figures are shown illustrating the optimal point-to-point movement of each individual agent.

Starting, in Section 7.1, with a brief introduction to multi-agent systems, Section 7.2 continues with describing the networked system considered in this chapter and gives an application-driven motivation for the subsequently addressed hybrid optimal control problem. In Section 7.3, against the background of a graph-based representation of the multi-agent system, the governing equations of the agents' dynamic behavior are stated. Finally, Section 7.4 presents the numerical

results for various different multi-agent configurations and in Section 7.5 this chapter is concluded by commenting on some interesting properties of the considered multi-agent dynamics.

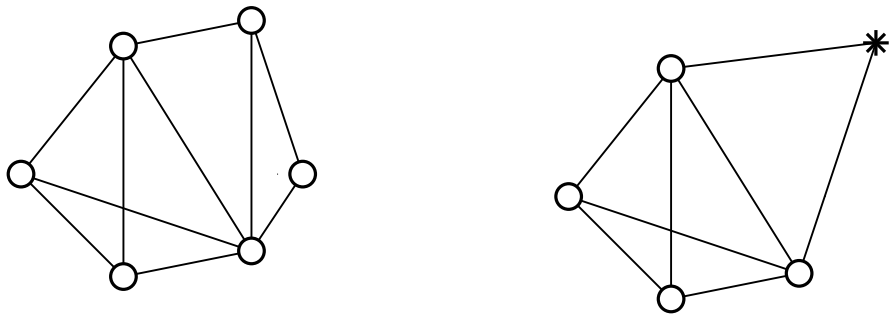
7.1 A Brief Introduction to Multi-Agent Systems

This section gives a brief overview of multi-agent systems, related work and applications.

Defining an *agent* as an autonomous entity equipped with a certain degree of sensing, processing, communication, and maneuvering capabilities, a *multi-agent system* is simply a collection of such agents, which have the ability to interact with their environment and also with other agents in the system. As a consequence, the dynamical behavior of such a multi-agent system depends not only on the dynamics of each individual agent, but also on the nature of their connection. A single link or connection between two elements may represent an information flow or a certain local interaction. What makes the multi-agent problem challenging is that the agents are subjected to limitations on the available information; that is, not every agent is directly connected to every other agent in the network.

In order to encode these limitations, graphs have been proved to be useful tools modeling the underlying communication topology. Among several important properties of such graph models, the graph Laplacian stands out, cf. [31,50], which is used in Section 7.3 to define the dynamics of the considered multi-agent system.

The numerous studies on networked systems [2,8,24,25,33–35,41,45–49,56,57, 65,67] are driven by a wide range of different applications including microsatellite clusters, autonomous underwater vehicles (AUVs), automated highway systems (AHSs), cooperative robot reconnaissance and manipulation, collaborative sensor arrays, distributed sensor networks (DSNs), terrestrial planet finder missions, formation flight control and the control of groups of unmanned vehicles. References on the mentioned topics are found in [25, 66].



(a) The Leaderless Approach (b) The Leader-Follower Approach

Figure 7.1: Examples for Networked Systems

7.2 Problem Statement, Motivation, and Applications

The optimal control problem under consideration, which is motivated by various practical applications, e.g., in the field of mobile robot navigation and distributed sensor networks, can be characterized as a hybrid point-to-point transfer problem, cf. Section 3.1; that is, the ultimate goal is to move a group of interconnected agents with *regional changing formation* optimally from a given start configuration to a fixed final destination.

Generally speaking, two distinctively different approaches have emerged in the rapidly expanding field of multi-agent systems and control. One field of investigation are homogeneous networks composed of numerous identical agents as shown in Figure 7.1(a). However, if any distinct agents are equipped with superior capabilities and allowed to take on a "leader" role, depicted by a star in Figure 7.1(b), the situation is called a heterogeneous formation. The latter case is considered in this chapter, where a collection of leaders dictate the motion of the followers. In particular, the followers' dynamics are governed by consensus-like local interaction rules, while the leaders are unconstrained in their movements. In fact, the leaders' trajectories are taken as the control input of the system.

Loosely speaking, one can think of the problem under investigation as the autonomous sheep-herding problem; in other words, the goal is to answer the

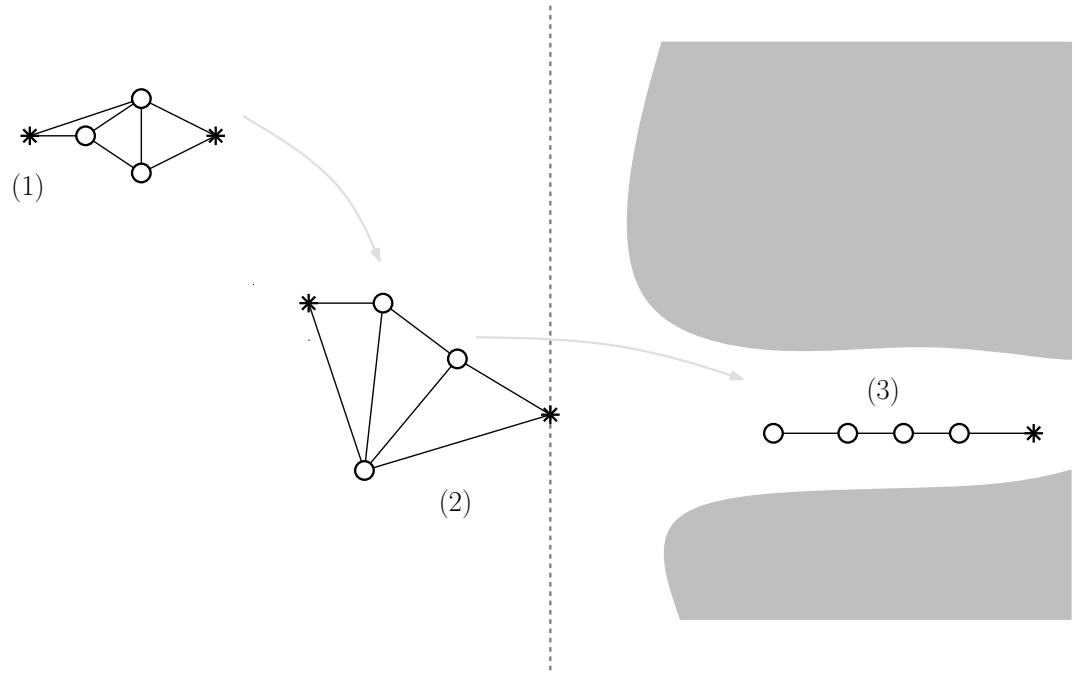


Figure 7.2: The Hybrid Point-To-Point Problem in the Case of a Multi-Agent System

question, "How should the herding dogs move in order to maneuver the herd in an desired optimal way to a given final destination?" This problem was previously considered in [35] and [8]. However, the novelty of the subsequent approach lies in the fact, that the agents' formation, i.e., the connections between the individual agents and also their assigned roles as leaders and followers, changes depending on the location of the agents. This switching behavior is shown in Figure 7.2, where a group of agents is moved from the start configuration (1) to the final arrangement (3) and a change in the formation of the network takes place as the first leader arrives at the switching boundary, situation (2). The time horizon for the entire motion from (1) to (3) is specified *a priori* and also an optimal control objective is given in advance. In fact, as can be seen in Section 7.3, where, finally, the equations governing the agents' dynamic behavior are presented, the illustrated multi-agent optimal control problem is simply a special case of the hybrid point-to-point problem previously considered in Chapter 5.

Nevertheless, notable applications are associated with the proposed multi-agent problem. As pointed out by [20], in the field of robotics, one can think of situations,

where robots should spread out first when navigating and exploring (unknown) free-space, while a more tight formation is to prefer when negotiating cluttered environments. This behavior is indicated in Figure 7.2, where a line formation is chosen as a reaction on the narrow path in the second region (on the right side of the switching line). From another point of view, sensor coverage problems may force the robots in some regions to stay closer together, where otherwise a wide-spread formation is more effective for attaining the goal. This idea of using different regional-dependent robot formations to get a better overall performance has lately received considerable attention, especially, in the investigations on using groups of mobile robots to tackle more advanced problems such as search-and-rescue and search-and-destroy tasks.

7.3 The Agents' Dynamic Behavior

A linear dynamic system models the proposed leader-follower behavior, where the leaders' positions are taken to be the system inputs and the followers' dynamics are given by decentralized averaging rules, i.e., their dynamics depend only on the relative displacements between the interacting agents.

Consider a group of N_A robots, whose positions $x_i = (x_{i,1}\ x_{i,2} \dots\ x_{i,n})'$, $i \in \{1, 2, \dots, N_A\}$ take on values in $\mathbb{R}^n$. Note that the definition of the variable x_k differs here from the one in Chapter 4 and Chapter 5. Moreover, it is assumed that the agent's dynamics in the different dimensions are decoupled. Hence, each dimension can be considered independently, and it is sufficient to analyze the performance along a single dimension. In other words, let $x_{i,k} \in \mathbb{R}$, $i \in \{1, 2, \dots, N_A\}$, be the position of the ith agent in dimension k, and let $\xi_k = (x_{1,k}\ x_{2,k}\ \dots\ x_{N_A,k})' \in \mathbb{R}^{N_A}$ be the aggregated state vector of the group of agents in direction $k \in \{1, 2, \dots, n\}$. For such a system, a widely adopted distributed control strategy is the so-called consensus equation, cf. [8],

$$\dot{x}_{i,k} = \sum_{j \in \mathcal{N}(i)} (x_{j,k} - x_{i,k}) \tag{7.1}$$

with $i \in \{1, 2, \ldots, N_A\}$ and $k \in \{1, 2, \ldots, n\}$. The set $\mathcal{N}(i)$ denotes the neighborhood of agent i and $j \in \mathcal{N}(i)$ encodes the fact that the information is allowed to flow from agent j to agent i; that is, there is a connection or communication link between agent j and agent i. Consequently, equation (7.1) represents a linear, time-invariant, decentralized control law, which explicitly represents the underlying network topology and the limited information of an individual agent. This nearest-neighbor rule (7.1) arises, especially, when dealing with agreement or rendezvous problems, which are concerned with finding decentralized strategies that achieve convergence to a common value. In fact, the followers' dynamics in the considered heterogeneous network is simply given by the above averaging rule (7.1), whereas the leader positions are taken as system inputs. However, the complete leader-follower dynamics are presented later in this section.

Throughout this chapter, the network topology is assumed to be static in each region D_i, $i \in \mathcal{I}$ of the state space $\mathcal{X}$, i.e., staying in one region, $\mathcal{N}(i)$ does not vary over time. In fact, the consensus equation (7.1) has been thoroughly studied for static as well as dynamic,i.e., time varying networks. A representative sample of some of the highlights in this area of research can be found in [19, 25, 33, 40, 41, 45, 46, 56, 65].

Graph theory can provide a variety of tools for analyzing such control strategies [29]. In the following, the communication topology is modeled as a graph $\mathcal{G} = (V, E)$, where the set of nodes $V = \{v_1, \ldots, v_{N_A}\}$ correspond to the different agents and the set of edges $E \subset V \times V$ to available inter-agent communication links. Note that $(v_i, v_j) = (v_j, v_i) \in E$, if and only if a communication link exists between agents i and j. Associated with the graph $\mathcal{G}$ are two distinctive matrix representations: the adjacency matrix $A(\mathcal{G})$ defined as

$$A(\mathcal{G}) = [a_{ij}] = \begin{cases} 1 & \text{if} \quad (v_i, v_j) \in E(\mathcal{G}), \\ 0 & \text{otherwise} \end{cases} \tag{7.2}$$

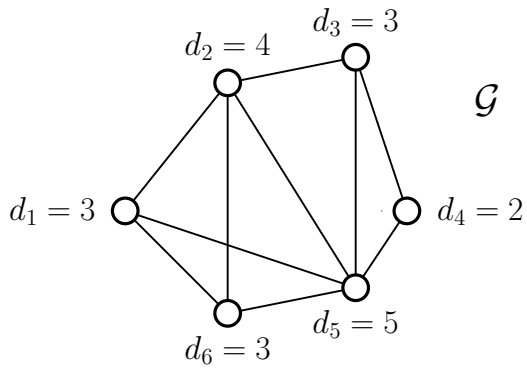


Figure 7.3: Illustration of the Vertex Degrees d_i for a Given Graph $\mathcal{G}$

and the degree matrix $D(\mathcal{G})$ given by

$$D(\mathcal{G}) = [d_{ij}] = \begin{cases} d_i & \text{if} \quad i = j, \\ 0 & \text{otherwise}, \end{cases} \tag{7.3}$$

where d_i denotes the degree of a graph vertex $v_i \in V$, i.e., the number of graph edges which touch v_i. Figure 7.3 illustrates the vertex degrees d_i corresponding to the vertices $v_i \in V$ of the graph $\mathcal{G} = (V, E)$.

Now, in order to relate the graph representation $\mathcal{G}$ of the underlying network topology with the consensus equation (7.1), the preliminary definitions, (7.2) and (7.3), are used to introduce a well-known and well-studied graph-theoretic concept, the graph Laplacian $\mathcal{L}(\mathcal{G})$, as

$$\mathcal{L}(\mathcal{G}) = D(\mathcal{G}) - A(\mathcal{G}). \tag{7.4}$$

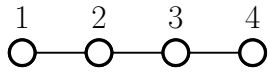
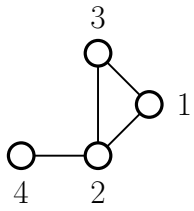
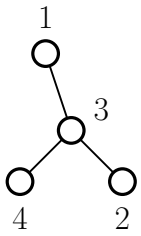
A number of interesting and outstanding properties are related with the matrix $\mathcal{L}(\mathcal{G})$, which are discussed in greater detail in [25, 29, 36] and the references herein. Finally, using the definition of the graph Laplacian $\mathcal{L}(\mathcal{G})$, equation (7.4), and the precise description of the neighborhood set

$$\mathcal{N}(i) = \{j \mid (v_i, v_j) \in E(\mathcal{G})\}, \tag{7.5}$$

the decentralized control law in (7.1) can be written as

$$\dot{\xi}_k = -\mathcal{L}(\mathcal{G})\, \xi_k, \tag{7.6}$$

Table 7.1: Three Different Network Topologies and Their Corresponding Graph Laplacians

	$\mathcal{L}^{\vert} = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$
	$\mathcal{L}^{\triangle} = \begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}$
	$\mathcal{L}^{\lambda} = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$

where $k \in \{1, 2, \ldots, n\}$. Note that, under the dynamics (7.6) and with the assumption of a connected graph $\mathcal{G}$, all agents approach asymptotically the same point independent of the given initial positions. For a detailed proof refer to [34] or [41]. In Table 7.1, three different graphs, namely, a line, triangle, and star formation, are depicted and their corresponding graph Laplacians, $\mathcal{L}^{\vert}$, $\mathcal{L}^{\triangle}$, and $\mathcal{L}^{\lambda}$, are given. These network structures are chosen later, in the examples of Section 7.4, as underlying communication topologies of the considered multi-agent systems.

As indicated at the beginning of this section, the dynamics of the considered heterogeneous multi-agent system are composed of two distinctive behaviors: The follower agents move autonomously based on local, consensus-like interaction rules and their governing equations are precisely given by (7.6), i.e., for a follower agent $f \in \{1, 2, \ldots, N_A\}$ the dynamics depend only on the positions of the other agents

in the system

$$\dot{x}_{f,k} = \sum_{j \in \mathcal{N}(f)} (x_{j,k} - x_{f,k}), \quad k \in \{1, 2, \ldots, n\}. \tag{7.7}$$

In contrast to the followers' dynamics, the leaders' velocities are assumed to be directly controllable, i.e., the dynamics of a leader agent $l \in \{1, 2, \ldots, N_A\}$ have the form

$$\dot{x}_{l,k} = u_{lk}, \qquad u_{lk} \in \mathbb{R}, \qquad k \in \{1, 2, \ldots, n\}. \tag{7.8}$$

Important to note is that the equations (7.7) and (7.8) contain the complete information on the dynamic behavior of the considered multi-agent system. However, in order to solve, in Section 7.4, the associated hybrid point-to-point problem, a more compact notation is preferable. In fact, after the definition of several new variables, the overall multi-agent dynamics can be written in the standard form $\dot{\xi} = A\xi + Bu$.

First of all, a leader vector $\kappa = (\kappa_1, \kappa_2, \ldots, \kappa_{N_A})' \in \mathbb{R}^{N_A}$ is introduced as a formal way of specifying which agent i is assigned with a leader role, i.e.,

$$\kappa_i = \begin{cases} 1 & \text{if the } i\text{th agent is a leader,} \\ 0 & \text{if the } i\text{th agent is a follower,} \end{cases} \tag{7.9}$$

where $i \in \{1, 2, \ldots, N_A\}$. The total number of leaders is denoted by $|\kappa|$. Moreover, based on equation (7.9) and the graph Laplacian $\mathcal{L}(\mathcal{G}) = [l_{ij}] \in \mathbb{R}^{N_A \times N_A}$, the matrix $\mathcal{L}_\kappa(\mathcal{G}) \in \mathbb{R}^{N_A \times N_A}$ is defined as

$$\mathcal{L}_\kappa(\mathcal{G}) = [\ell_{ij}] = \begin{cases} l_{ij} & \text{if the } \kappa_i = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{7.10}$$

Finally, with the definition of the non-square matrix $B_\kappa \in \mathbb{R}^{N_A \times |\kappa|}$ as

$$B_\kappa = [b_{ij}] = \begin{cases} 1 & \text{if the } j\text{th non-zero entry in } \kappa \text{ is } \kappa_i, \\ 0 & \text{otherwise,} \end{cases} \tag{7.11}$$

the linear dynamic behavior of the overall system is given by

$$\dot{\xi}_j = \Big(-\mathcal{L}(\mathcal{G}) + \mathcal{L}_\kappa(\mathcal{G}) \Big) \xi_j + B_\kappa u_j \tag{7.12}$$

$$= A_\kappa(\mathcal{G})\, \xi_j + B_\kappa u_j, \tag{7.13}$$

where $A_\kappa(\mathcal{G}) \in \mathbb{R}^{N_A \times N_A}$ and $u_k \in \mathbb{R}^{|\kappa|}$. Note that equation (7.13) holds for each direction j separately, pointing out, again, that the individual directions $j \in \{1, 2, \ldots, n\}$ are completely decoupled. Nevertheless, in order to compute, in Section 7.13, the solution to the ybrid point-to-pont transfer problem, a compact mathematical representation of the overall multi-agent dynamics with the state vector $\xi = (\xi_1\ \xi_2\ \ldots\ \xi_n)' \in \mathbb{R}^{n \cdot N_A}$ and the input $u = (u_1\ u_2\ \ldots\ u_n)' \in \mathbb{R}^{n \cdot |\kappa|}$ is preferred, which is uniquely given by equation (7.13) and can be written in the standard form for linear dynamic systems

$$\dot{\xi} = A(\mathcal{G})\,\xi + B\,u, \tag{7.14}$$

where $A(\mathcal{G}) = diag(A_\kappa(\mathcal{G})) \in \mathbb{R}^{(n \cdot N_A)}$ and $B = diag(B_\kappa) \in \mathbb{R}^{(n \cdot N_A) \times (n \cdot |\kappa|)}$ are block diagonal matrices.

With relation to the following section, where the hybrid point-to-point problem is solved for the presented multi-agent systems with dynamics (7.14), two remarkable comments are made: First, the dimension of the input u depends on the number of leaders in the network. Consequently, in a multiregional framework, cf. Section 5.2, either the number of leaders must be the same in each region or the cost function (5.19) must satisfy some additional constraints. A different number of leaders in the different regions $D_i,\ i \in \mathcal{I}$ is, for example, possible, in the case of an input-independent cost $\ell(x(t), u(t)) = \ell(x(t))$ or in the case of assigning individual cost functions $\ell_i(x(t), u(t)),\ i \in \mathcal{I}$ to each region D_i as highlighted in Section 6.2.4. In addition, because of the linearity of overall multi-agent dynamics (7.14), the computational procedure, solving the hybrid multi-agent point-to-point problem numerically, is exactly the same as in the previous examples of Section 2.3.2, Section 4.6, and Section 5.8. In particular, all subsequent multi-agent examples are chosen in a way satisfying Assumption 5.1.

7.4 An Optimal Control Approach

This section definitely represents one of the highlights in this work, in that it presents numerical and graphical results, based on the hierarchical Dynamic Programming approach presented in Section 5, for the application-driven multi-agent hybrid point-to-point problem.

In the following, the dynamic behavior of a group of four agents, interacting through different network topologies, is studied, where the individual robots "live" in the two-dimensional space, i.e., $x_i \in \mathbb{R}^2,\ i \in \{1, 2, , 3, 4\}$. Furthermore, basically two different regional frameworks are taken into account: Section 7.4.1 starts with solving the point-to-point problem in one region. This is done, especially, in order to illustrate the influence of different network structures on the multi-agent dynamic behavior subjected to an optimal control policy. Later, in Section 7.4.2, these results are compared to the agents' optimal trajectories in a bimodal geometric framework. In both sections, a time horizon $T = 6$ is assumed for the transition between the initial and final configuration, ξ_0 and ξ_T, respectively. The cost function under consideration is given in the form

$$\ell(x(t), u(t)) = x'(t)Qx(t) + u'(t)Ru(t), \tag{7.15}$$

where $R = 2\, I_{(2\cdot|\kappa|)\times(2\cdot|\kappa|)}$ in all examples and $Q = I_{8\times 8}$ except for Example 7.2 and Example 7.6, where $Q = 0 \cdot I_{8\times 8}$. Note that the dynamics of a multi-agent system is uniquely specified by the leader vector κ and the graph Laplacian $\mathcal{L}(\mathcal{G})$, where the following examples study, in particular, the network structures depicted in Table 7.1.

7.4.1 The Point-To-Point Problem in One Region

The main focus of this section is to illustrate, by solving the (non-hybrid) multi-agent point-to-point problem in one region, how the variation of the boundary conditions and the communication topology influences the dynamic behavior of the multi-agent system under an optimal control policy. In particular, the topologies presented in Table 7.1 are considered. As a result of the linear model for the

agents' dynamic behavior, the approach of Section 2.3.2 applies similarly to the actual multi-agent problem and the computational procedure stays exactly the same. In addition, especially noteworthy is that the following non-hybrid optimal trajectories serve later, in Section 7.4.2, where the optimal multi-agent trajectories are calculated in a bimodal framework, as interesting results to compare with.

Example 7.1. In this example, the multi-agent system (7.14) is driven from the start configuration

$$\xi_0 = \begin{pmatrix} -1 & 0 & -1.5 & 0.5 & -1.5 & -0.5 & -2 & 0.5 \end{pmatrix}' \tag{7.16}$$

and the final positions

$$\xi_T = \begin{pmatrix} 6 & 1 & 5 & 1 & 4 & 1 & 3 & 1 \end{pmatrix}'. \tag{7.17}$$

The leader role is assigned to agent 1 and 4, i.e.,

$$\kappa = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}'. \tag{7.18}$$

The agents' optimal behavior is considered for two different network topologies, the line formation with $\mathcal{L}_|$ and the triangle formation with $\mathcal{L}^\triangle$, cf. Table 7.1. The comparison of both results, depicted for the line formation in Figure 7.4 and Figure 7.5 and for the triangle formation in Figure 7.6 and Figure 7.7, shows the influence of the network structure on the optimal trajectories and allows interesting conclusions on the characteristics of the different formations. The distinguished role of the leaders is expressed in Figure 7.4 and Figure 7.6 by a star marker, whereas the followers are displayed by circles. The cost associated with the optimal trajectories takes, in the case of the line formation, the value $J^*_| \approx 226.01$ and the value $J^*_\triangle \approx 594.95$ in the case of the triangle formation. □

Similar to Example 7.1, the following example illustrates the group's behavior in a line formation, $\mathcal{L}(\mathcal{G}) = \mathcal{L}^|$, in comparison to the dynamics in a star formation, $\mathcal{L}(\mathcal{G}) = \mathcal{L}^\lambda$, cf. Table 7.1.

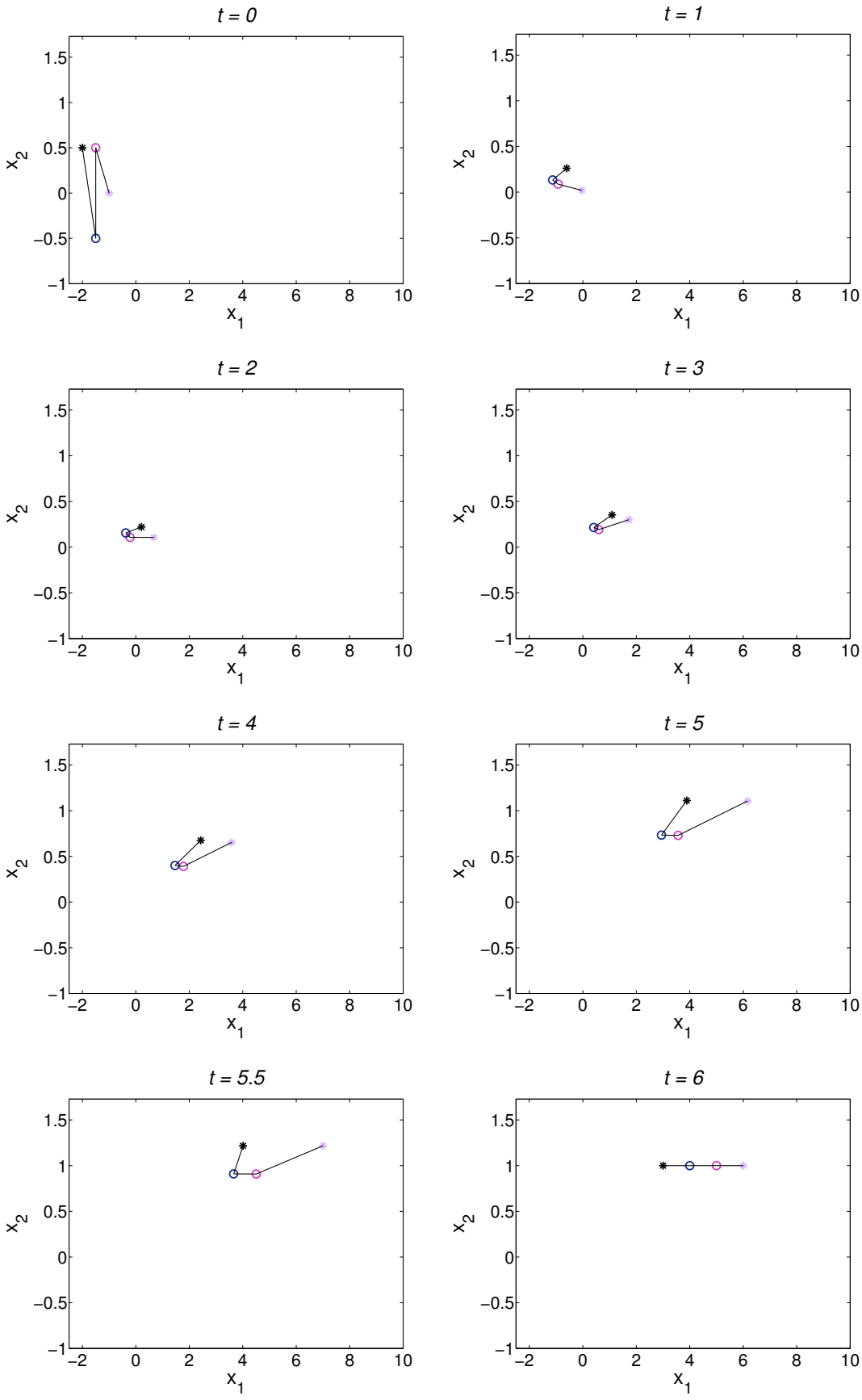
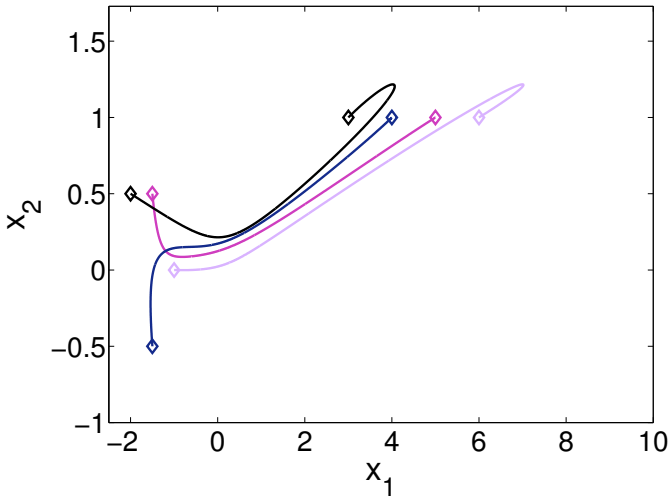
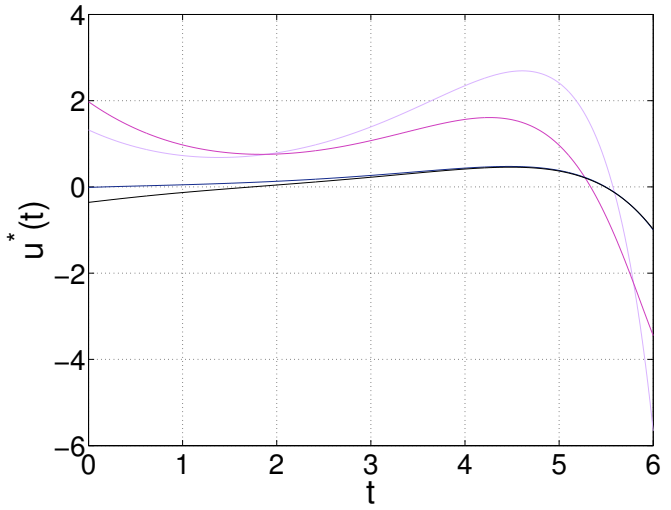


Figure 7.4: Optimal Group Behavior Corresponding To Example 7.1; Line Formation $\mathcal{L}(\mathcal{G}) = \mathcal{L}^{|}$



(a) Optimal States $(x^*_{k,1}(\cdot), x^*_{k,2}(\cdot))$ for each agent k, $k \in \{1, 2, 3, 4\}$



(b) Optimal Input $u^*(\cdot)$

Figure 7.5: Optimal Trajectories Corresponding To Example 7.1; Line Formation $\mathcal{L}(\mathcal{G}) = \mathcal{L}^{|}$

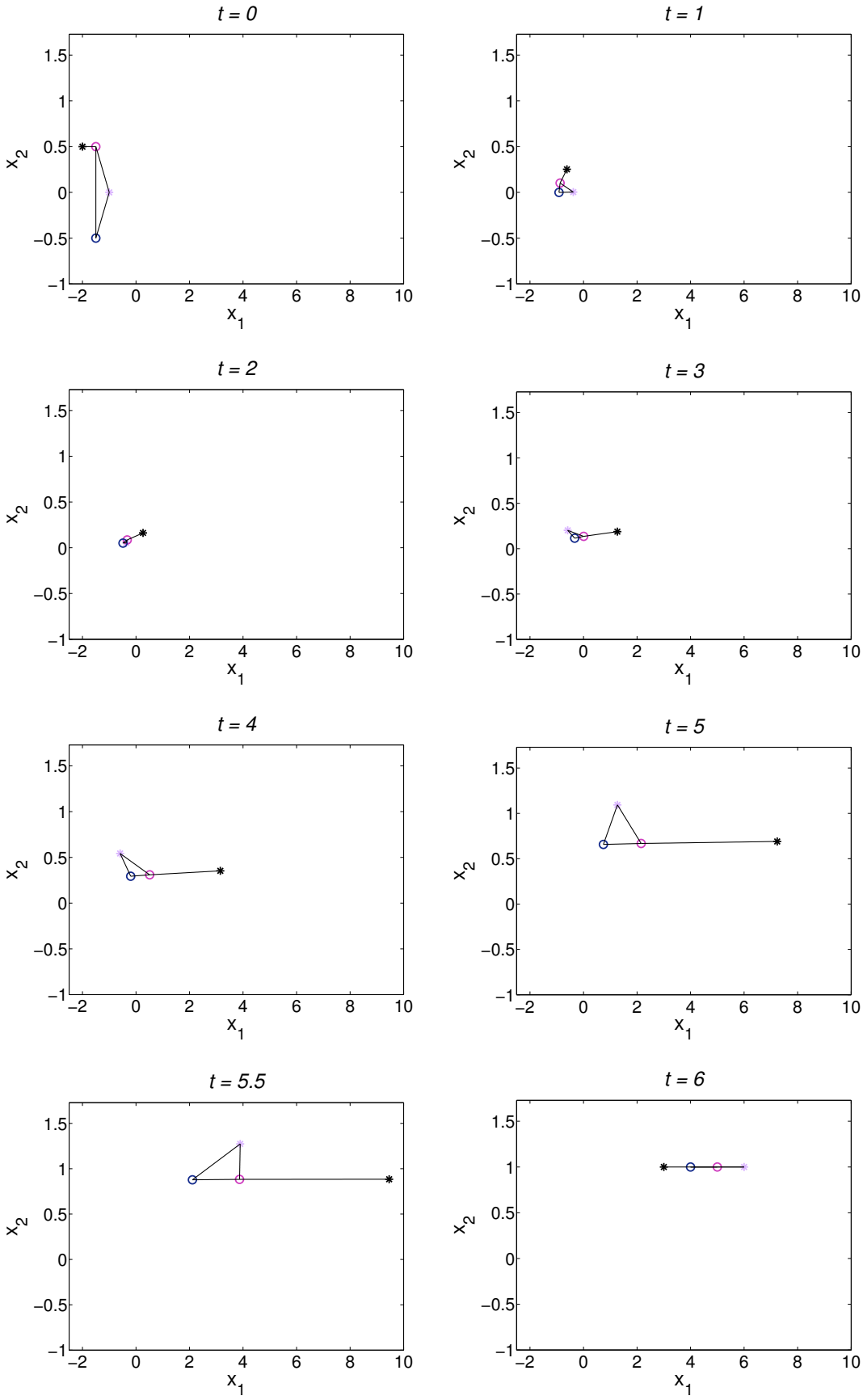
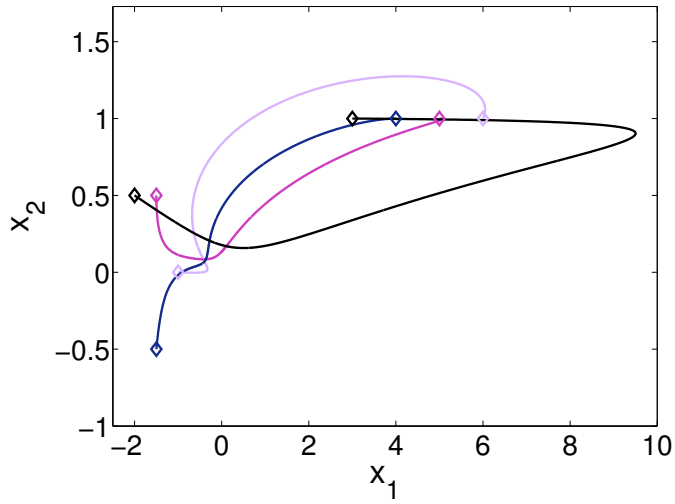
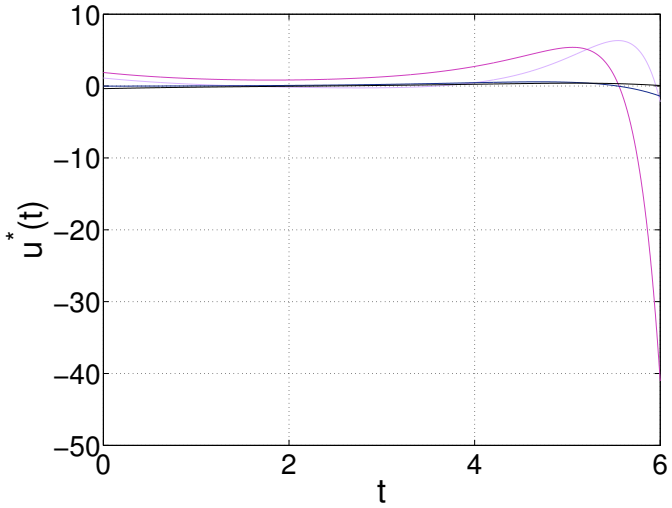


Figure 7.6: Optimal Group Behavior Corresponding To Example 7.1; Triangle Formation $\mathcal{L}(\mathcal{G}) = \mathcal{L}^{\triangle}$



(a) Optimal States $(x^*_{k,1}(\cdot), x^*_{k,2}(\cdot))$ for each agent k, $k \in \{1, 2, 3, 4\}$



(b) Optimal Input $u^*(\cdot)$

Figure 7.7: Optimal Trajectories Corresponding To Example 7.1; Triangle Formation $\mathcal{L}(\mathcal{G}) = \mathcal{L}^{\triangle}$

Example 7.2. Another set of boundary conditions,

$$\xi_0 = \begin{pmatrix} 0.1 & -0.1 & -1 & -0.5 & -1 & 0 & -1.5 & 0.5 \end{pmatrix}' \tag{7.19}$$

and

$$\xi_T = \begin{pmatrix} 6 & 0 & 5.5 & 0 & 5 & 0 & 4.5 & 0 \end{pmatrix}', \tag{7.20}$$

is considered in this example and, again, the multi-agent behavior for two different communication topologies are compared, where three leaders

$$\kappa = \begin{pmatrix} 1 & 1 & 0 & 1 \end{pmatrix}' \tag{7.21}$$

are assumed. The optimal trajectories associated with the line formation, $\mathcal{L}(\mathcal{G}) = \mathcal{L}^{|}$, are presented in Figure 7.8 and Figure 7.9, whereas the agents organized in a star formation, $\mathcal{L}(\mathcal{G}) = \mathcal{L}^{\lambda}$, show an optimal behavior as depicted in Figure 7.10 and Figure 7.11. The optimal cost for the considered situations is given by $J^*_{|} \approx 43.06$ in the case of the line structure and $J^*_{\lambda} \approx 37.86$ in the case of the star network. Consequently, for the previously specified cost function and the given initial and final states, (7.19) and (7.20), the star formation is to prefer. Furthermore, an interesting comparison can be made between the point-to-point solution for the line network in Example 7.1, where the boundary conditions are given by (7.16) and (7.17), and the optimal solution for the line formation presented in this example.

□

The above examples, representing the solution to the non-hybrid point-to-point problem, are now compared with the optimal solution in a bimodal framework. In particular, in the subsequent examples, Example 7.5 and Example 7.6, exactly the same boundary conditions, ξ_0 and ξ_T, are assumed as in Example 7.5 and Example 7.6, respectively; however, the network topology changes depending on a given switching rule and, of course, influences the optimal paths of the individual agents.

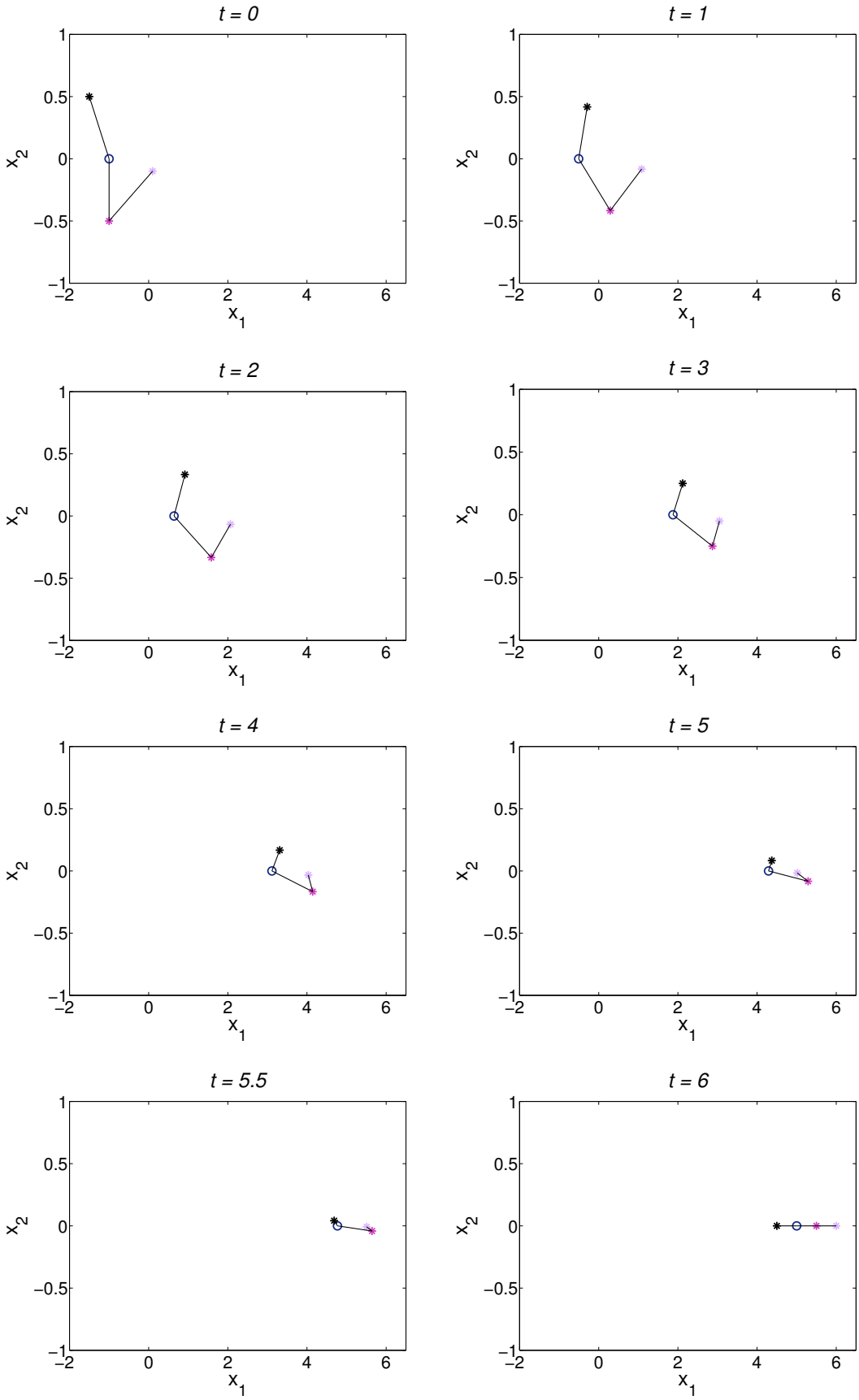
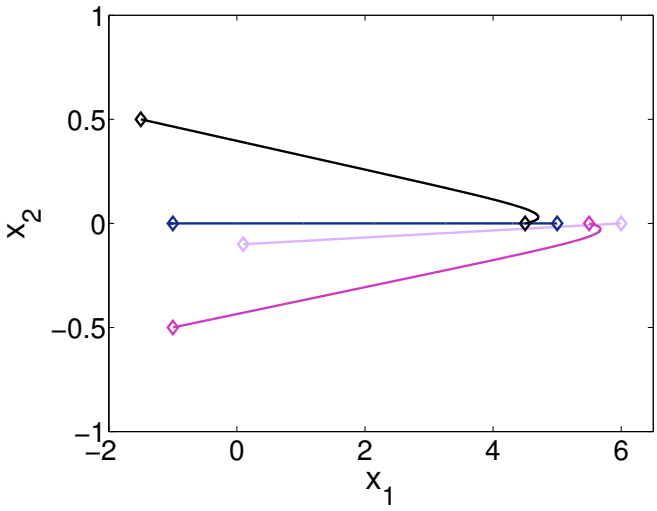
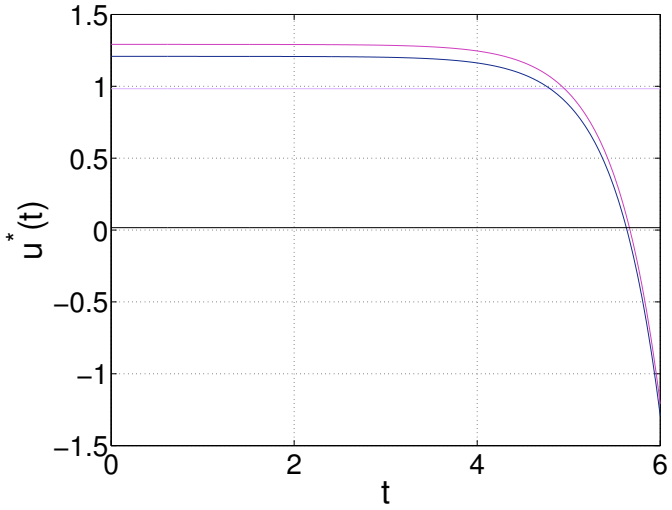


Figure 7.8: Optimal Group Behavior Corresponding To Example 7.2; Line Formation $\mathcal{L}(\mathcal{G}) = \mathcal{L}^{|}$



(a) Optimal States $(x^*_{k,1}(\cdot), x^*_{k,2}(\cdot))$ for each agent k, $k \in \{1, 2, 3, 4\}$



(b) Optimal Input $u^*(\cdot)$

Figure 7.9: Optimal Trajectories Corresponding To Example 7.2; Line Formation $\mathcal{L}(\mathcal{G}) = \mathcal{L}^{|}$

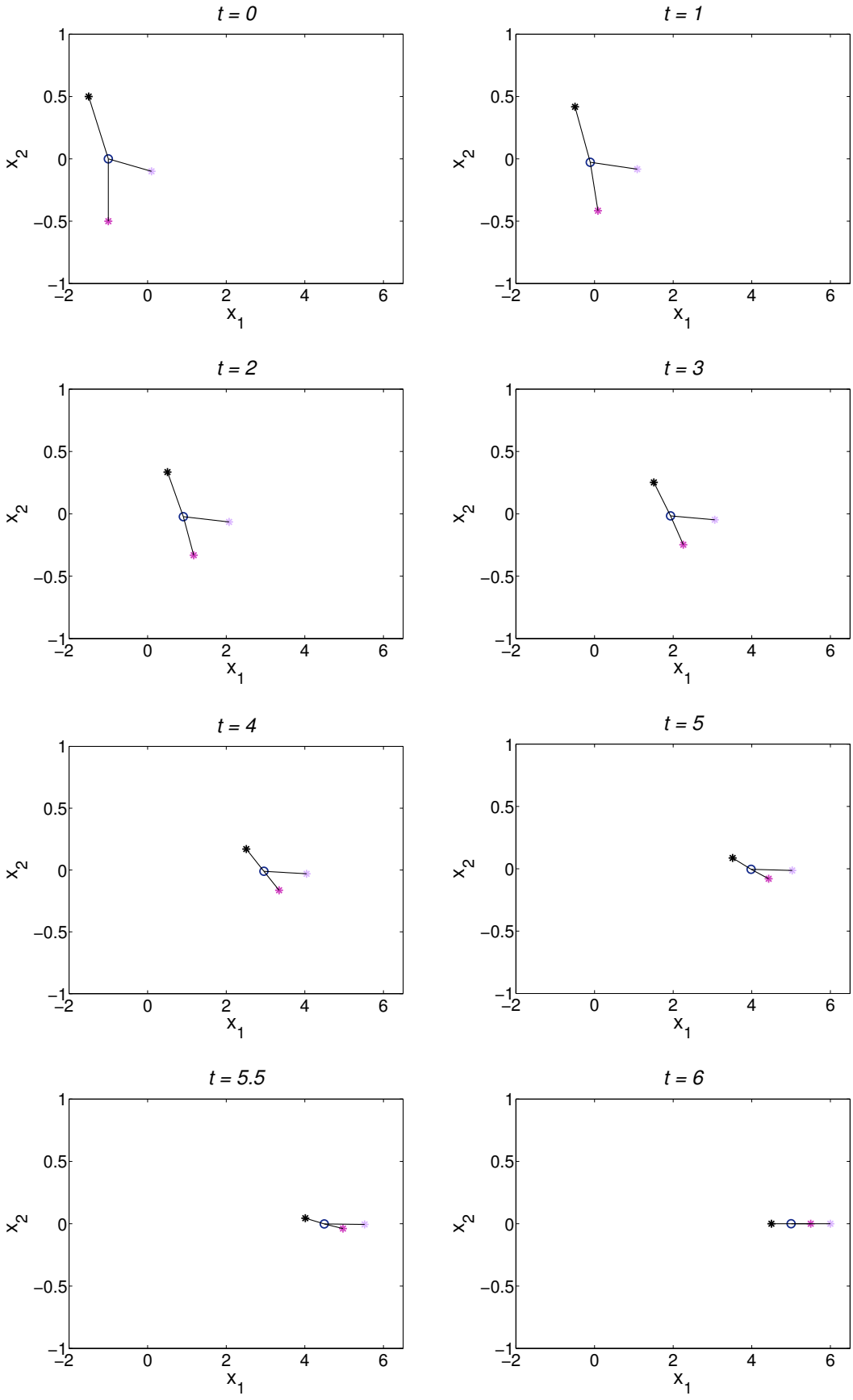
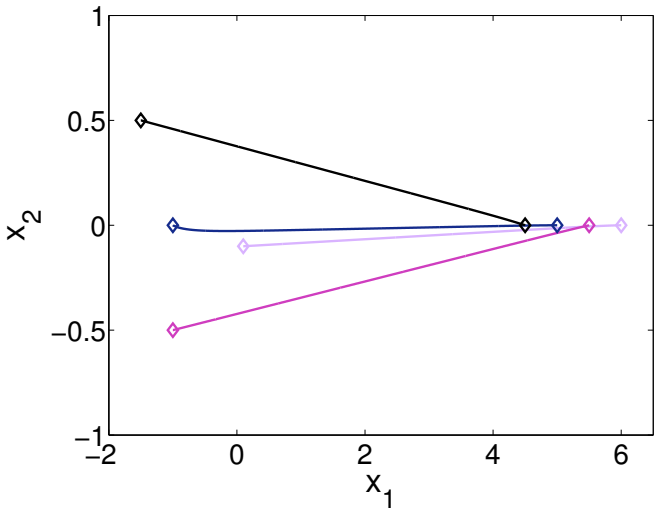
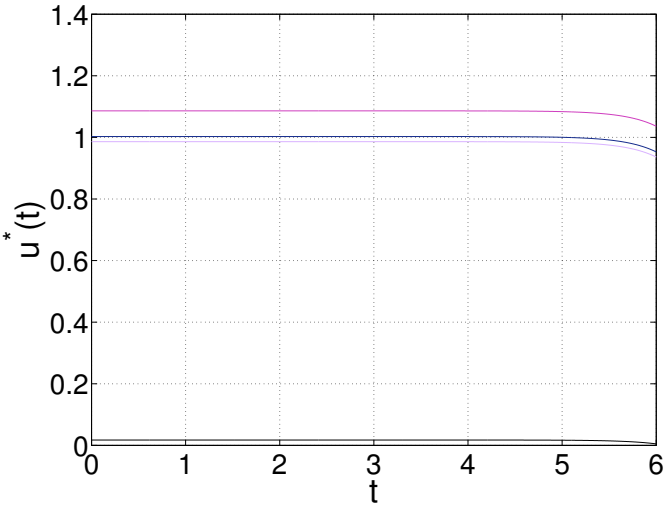


Figure 7.10: Optimal Group Behavior Corresponding To Example 7.2; Star Formation $\mathcal{L}(\mathcal{G}) = \mathcal{L}^{\lambda}$



(a) Optimal States $(x^*_{k,1}(\cdot), x^*_{k,2}(\cdot))$ for each agent k, $k \in \{1, 2, 3, 4\}$



(b) Optimal Input $u^*(\cdot)$

Figure 7.11: Optimal Trajectories Corresponding To Example 7.2; Star Formation $\mathcal{L}(\mathcal{G}) = \mathcal{L}^\lambda$

7.4.2 The Bimodal Point-To-Point Problem

After the insightful examples in Section 7.4.1, illustrating the optimally controlled behavior of the proposed leader-follower dynamics (7.14) under varying boundary conditions and network topologies, this section, finally, investigates the influence of a regional-dependent switch in the agents' formation, caused either by a changed role assignment or by a different interconnection structure. Compelling results are obtained for both cases. Because of the linear multi-agent model (7.14), the approach proposed in Section 4.5 and Section 4.6 solves also the following multi-agent problems, where the examples are chosen in a way that Assumption 5.1 is satisfied.

First, some common properties, which hold for all subsequent examples, are briefly stated. Considered is a state space $\mathcal{X}$, where $\mathcal{X} \subseteq \mathbb{R}^8$ in the case of four agents $x_k \in \mathbb{R}^2,\ k \in \{1,2,3,4\}$, which is divided into the two regions

$$D_1 = \left\{\xi \mid \begin{pmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{pmatrix}\xi < 1\right\} \quad \text{and} \tag{7.22}$$

$$D_2 = \left\{\xi \mid \begin{pmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{pmatrix}\xi > 1\right\}. \tag{7.23}$$

Since a switching manifold of the form $c'\,\xi = d$ is a linear combination of all states $x_{k,i},\ i \in \{1, 2\}$ of every agents $k \in \{1, 2, 3, 4\}$, the interpretation and illustration can be very difficult. Nevertheless, in the above case, a switch takes simply place when the x_1-component of agent 1 reaches the value 1. The transition behavior of the regional dynamics system is determined by the automaton shown in Figure 5.6. However, in all subsequent examples, the upper bound on the total number of switches along a hybrid trajectory is given by $N = 1$. Furthermore, the starting configuration lies always in region D_1, i.e. $\xi_0 \in D_1$, and the final destination satisfies $\xi_T \in D_2$. The numerical solution is obtained by discretizing the time interval $[0, 6]$ into 6 equally spaced temporal steps and by discretizing the switching manifold

$$m_{(1,2)}: \begin{pmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{pmatrix}\xi = 1 \tag{7.24}$$

into 2 equally spaced spatial steps over the interval $[-0.5, 0.5]$ for each dimension $x_{k,i}$, $i \in \{1, 2\}$ and $k \in \{1, 2, 3, 4\}$, except for $x_{1,1}$, which is specified as $x_{1,1} = 1$ by (7.24). In the following, the system specifications, that is, the topologies $\mathcal{L}_1$ and $\mathcal{L}_2$ associated with the regions D_1 and D_2, the corresponding leader vectors κ_1 and κ_2, and the initial and final constraint, $\xi(0) = \xi_0$ and $\xi(T) = \xi_T$, are briefly mentioned at the beginning of each example. In addition, the optimal cost $W^1(\xi_0, q_1, \xi_T, q_1, T)$ is given. Special emphasis is, however, place on an interpretation and evaluation of the results.

The first example illustrates the case, where a formation switch simply means a change of the leader and follower roles. The interconnections do not change during the whole maneuver.

Example 7.3. For the agents' initial and final position,

$$\xi_0 = \begin{pmatrix} -1 & 0 & -1.5 & 0 & -2 & 0 & -2.5 & 0 \end{pmatrix}' \tag{7.25}$$

and

$$\xi_T = \begin{pmatrix} 3 & 1 & 4 & 0.5 & 4 & -0.5 & 3 & -1 \end{pmatrix}', \tag{7.26}$$

the leader vectors,

$$\kappa_1 = \begin{pmatrix} 0 & 1 & 1 & 0 \end{pmatrix}' \tag{7.27}$$

and

$$\kappa_2 = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}', \tag{7.28}$$

and the formations $\mathcal{L}_1 = \mathcal{L}_2 = \mathcal{L}^{|}$, the resulting hybrid optimal trajectories are depicted in Figure 7.12, Figure 7.13, and Figure 7.14 and the associated value function is given by $W^1(\xi_0, q_1, \xi_T, q_1, T) \approx 259.18$. The circles in Figure 7.13 display the individual states of the agents at the switching time $t = 3$. The dashed line in Figure 7.14 represents the optimal discrete state $q^*(\cdot)$, where the lower level corresponds to $q(t) = q_1$ and the higher level to $q(t) = q_2$.

Interesting in Figure 7.13, is the horizontal movement of all agents in region D_1 (left)which is certainly influenced by the chosen line formation. □

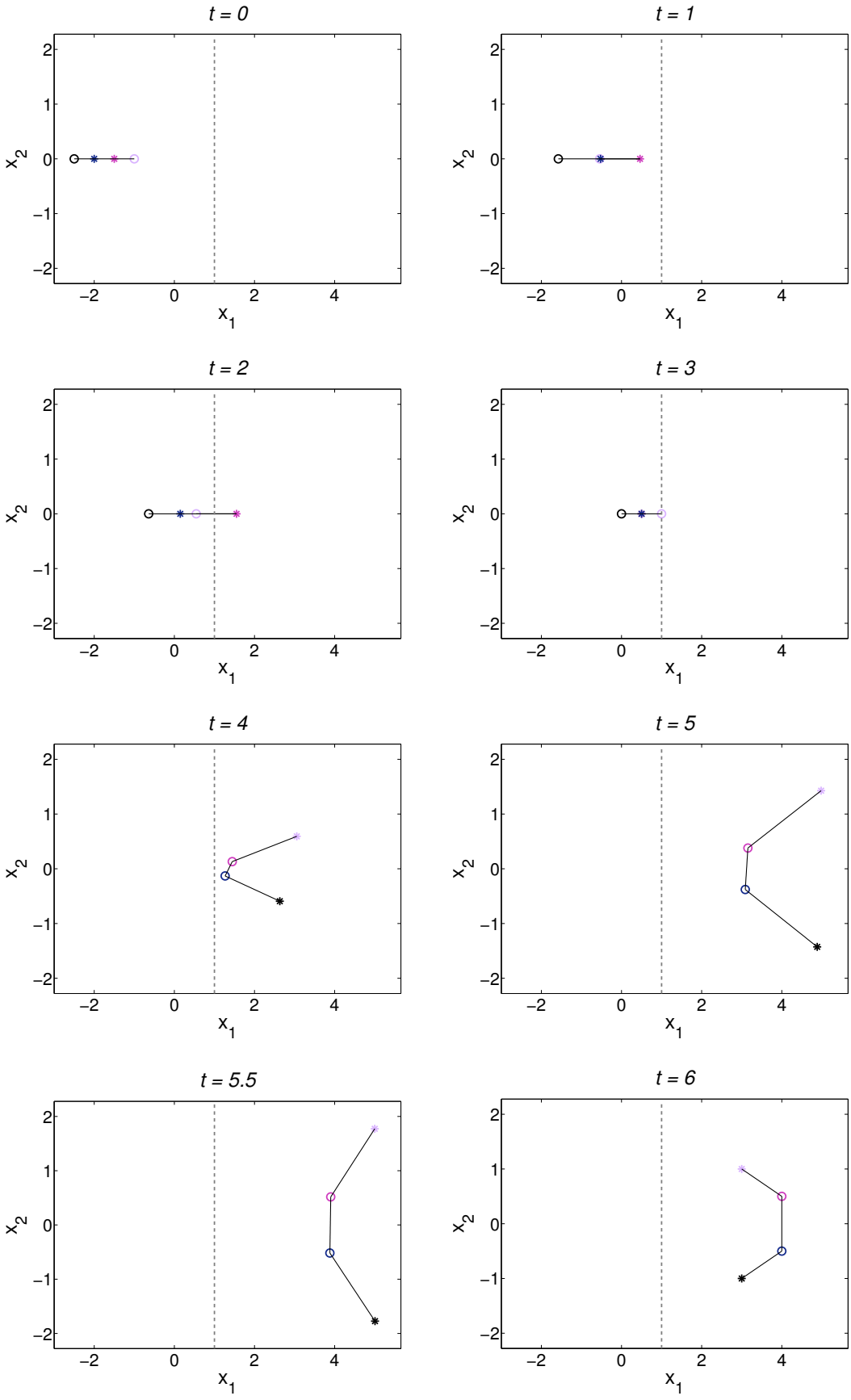


Figure 7.12: Optimal Group Behavior Corresponding To Example 7.3

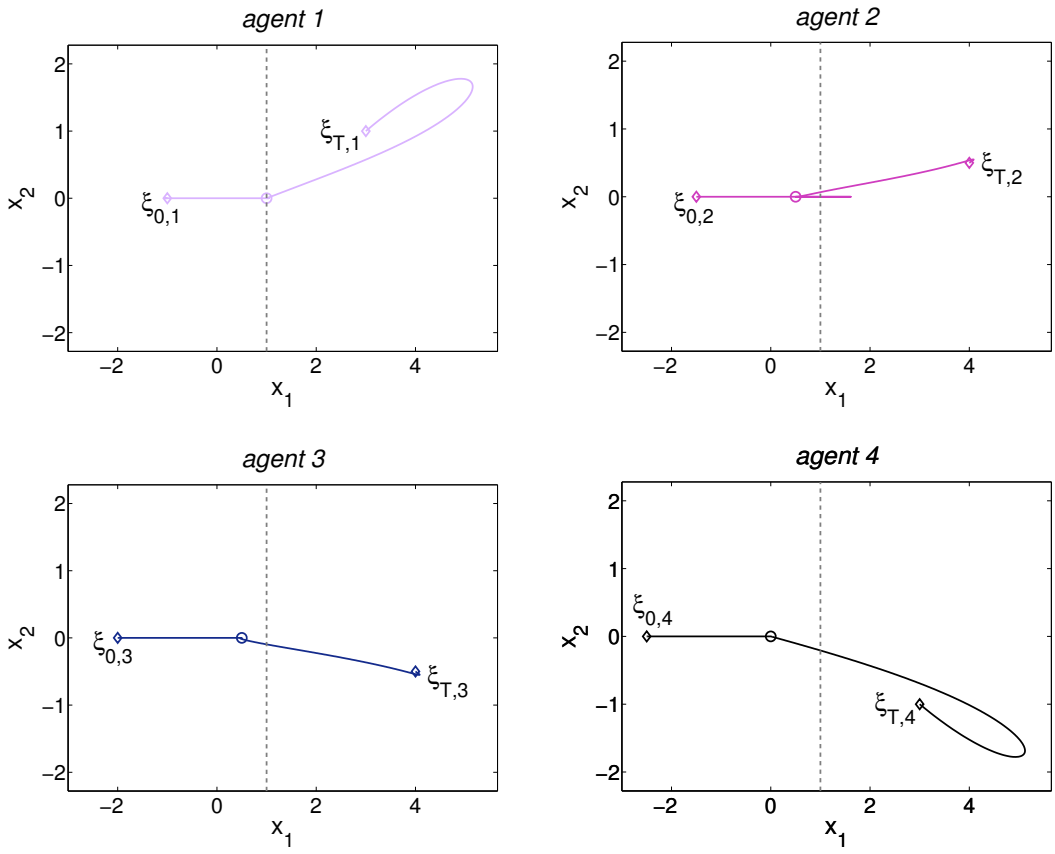


Figure 7.13: Optimal States $(x^*_{k,1}(\cdot),\, x^*_{k,2}(\cdot))$ Corresponding To Example 7.3 depicted separately for each agent k, $k \in \{1, 2, 3, 4\}$

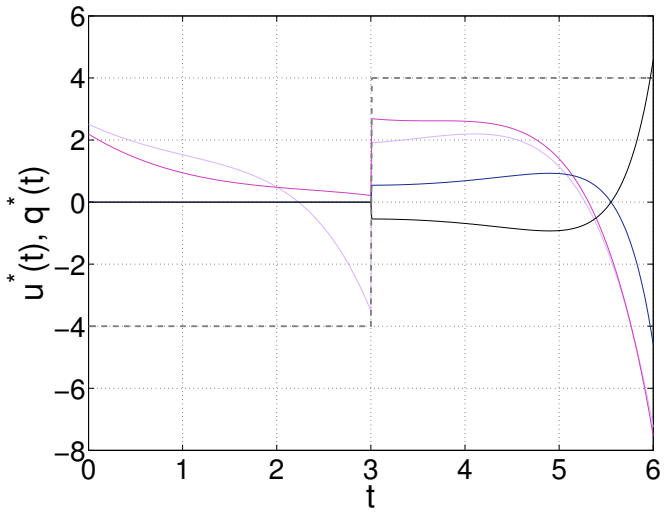


Figure 7.14: Optimal Input $u^*(\cdot)$ Corresponding To Example 7.3

In contrast to the previous example, in the subsequent problem statement the leader stay the same when crossing the switching manifold; however, the network structure changes what allows interesting conclusions on the characteristics of different topologies.

Example 7.4. The specifications on the boundary conditions,

$$\xi_0 = \begin{pmatrix} -1 & 0 & -1.5 & 0 & -2 & 0 & -2.5 & 0 \end{pmatrix}' \tag{7.29}$$

and

$$\xi_T = \begin{pmatrix} 6 & 1.5 & 5.5 & 1.25 & 5 & 1 & 4.5 & 0.75 \end{pmatrix}', \tag{7.30}$$

the leader vectors,

$$\kappa_1 = \kappa_2 = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}', \tag{7.31}$$

and the formations, $\mathcal{L}_1 = \mathcal{L}^{\triangle}$ and $\mathcal{L}_2 = \mathcal{L}^{|}$, result in the hybrid optimal trajectories displayed in Figure 7.15, Figure 7.16, and Figure 7.17 and the associated cost takes the value $W^1(\xi_0, q_1, \xi_T, q_1, T) \approx 291.48$.

Figure 7.15 is a perfect starting point to discuss the influence of the formation, but also of the optimal control objective, on the agents' group behavior. In region D_1 (left side), the strong connection between the agents, combined with the previously specified matrix $Q = I_{8\times 8}$ in the cost function (7.15), yield to a "shrinking" behavior, i.e., the agents move together and reach almost one point. After the switch at time $t = 2$, the line structure allows a re-opening of the formation. □

Finally, the following two examples, Example 7.5 and Example 7.6, show exactly the same boundary conditions and leader vectors $\kappa_1 = \kappa_2 = \kappa$ as the non-hybrid examples, Example 7.1 and Example 7.2, respectively. However, they combine the formations, separately considered in Section 7.4.1, to a bimodal framework. Hence, a comparison of the optimal trajectories and the value functions in Example 7.1 and Example 7.2 with the ones in Example 7.5 and Example 7.6, respectively, is particularly interesting.

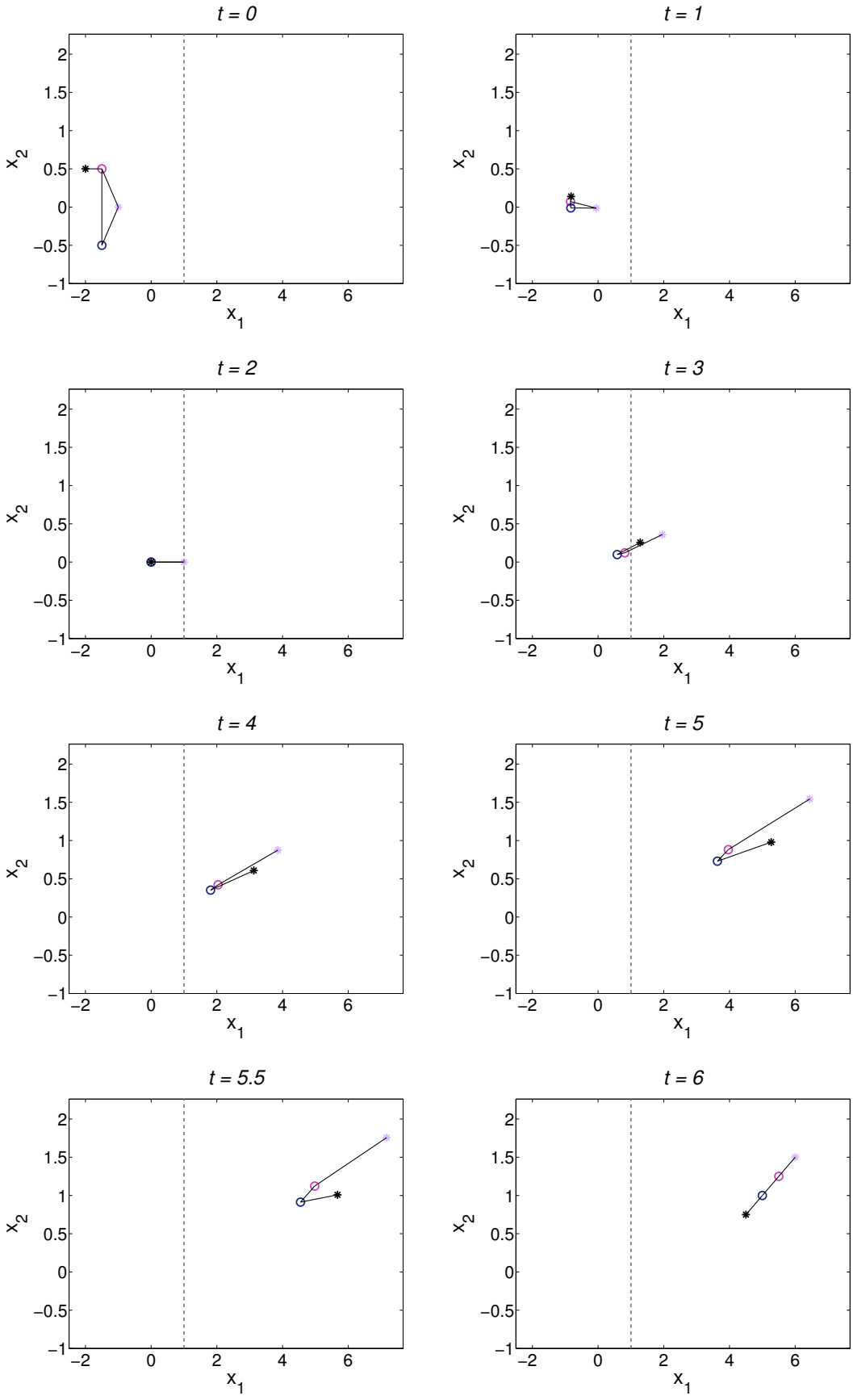


Figure 7.15: Optimal Group Behavior Corresponding To Example 7.4

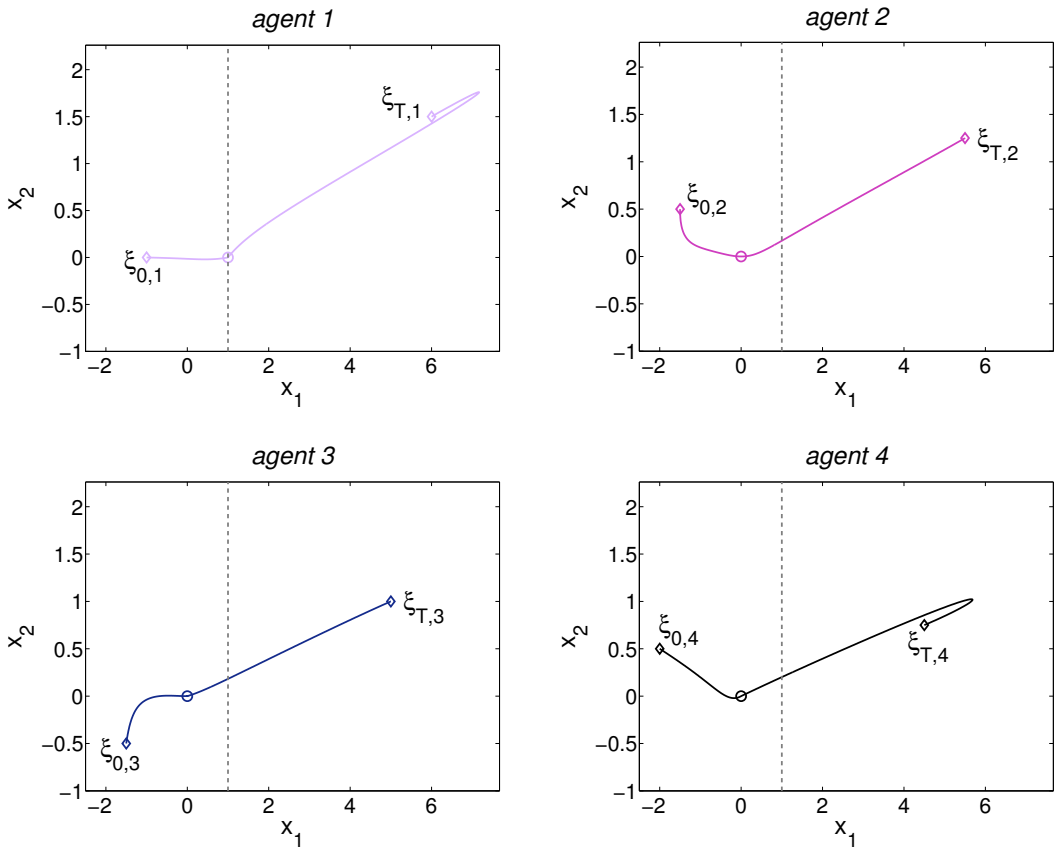


Figure 7.16: Optimal States $(x^*_{k,1}(\cdot), x^*_{k,2}(\cdot))$ Corresponding To Example 7.4 depicted separately for each agent k, $k \in \{1, 2, 3, 4\}$

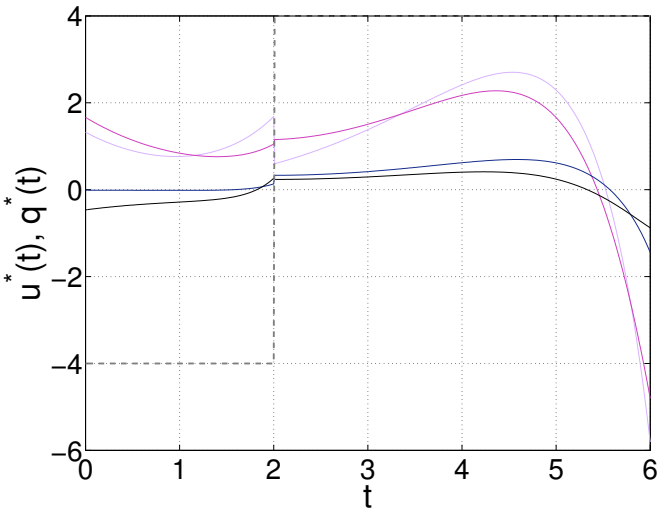


Figure 7.17: Optimal Input $u^*(\cdot)$ Corresponding To Example 7.4

Example 7.5. Here, the triangle formation $L^{\triangle}$ is associated with region D_1, i.e. $\mathcal{L}_1 = \mathcal{L}^{\triangle}$, and a line network $\mathcal{L}^{|}$ is chosen in region D_2, i.e., $\mathcal{L}_2 = \mathcal{L}^{|}$. For this geometric framework, the optimal cost is given by $W^1(\xi_0, q_1, \xi_T, q_1, T) \approx 225.84$. Compared with the costs in Example 7.1, it follows that, interestingly, the bimodal geometric structure yields to the lowest cost. □

Example 7.6. This example assumes $\mathcal{L}_1 = \mathcal{L}^{\lambda}$ and $\mathcal{L}_2 = \mathcal{L}^{|}$, which results in the value function $W^1(\xi_0, q_1, \xi_T, q_1, T) \approx 43.46$. In fact, compared with $J^*_{|}$ and $J^*_{\triangle}$ of Example 7.2, the one-regional case with triangle formation is the most efficient. □

7.5 Conclusions

With the aim of completing and concluding the general investigations on regional dynamics systems with a compelling application-driven example, this chapter introduces a heterogeneous multi-agent system, where a directly controlled group of leader agents dictates the followers' motion, which is given by local, consensus-like interaction rules. Based on a graph representation of the interconnections between the individual agents, the overall group dynamics reveals themselves as being a simple standard linear dynamic system. By solving the hybrid point-to-point problem for these kind of systems, not only the computability and applicability of our hierarchic approach is proven, but also the behavioral characteristics of the proposed agents' interaction dynamics are highlighted.

Compared to the examples previously addressed in Section 2.3.2, Section 4.6, and Section 5.8, the dimensionality of the multi-agent problem is significantly larger and increases, of course, with a growing number of agents. Computationally, this means an exponential increase in complexity known as the "curse of dimensionality" and previously explained in Section 2.2.3, Section 4.5 and Section 5.7. However, if the agent dynamics in each direction k, $k \in \{1, 2, \ldots, n\}$, where $x_{i,k} \in \mathbb{R}$ and $i \in \{1, 2, \ldots, N_A\}$, are independent and also a decoupled cost

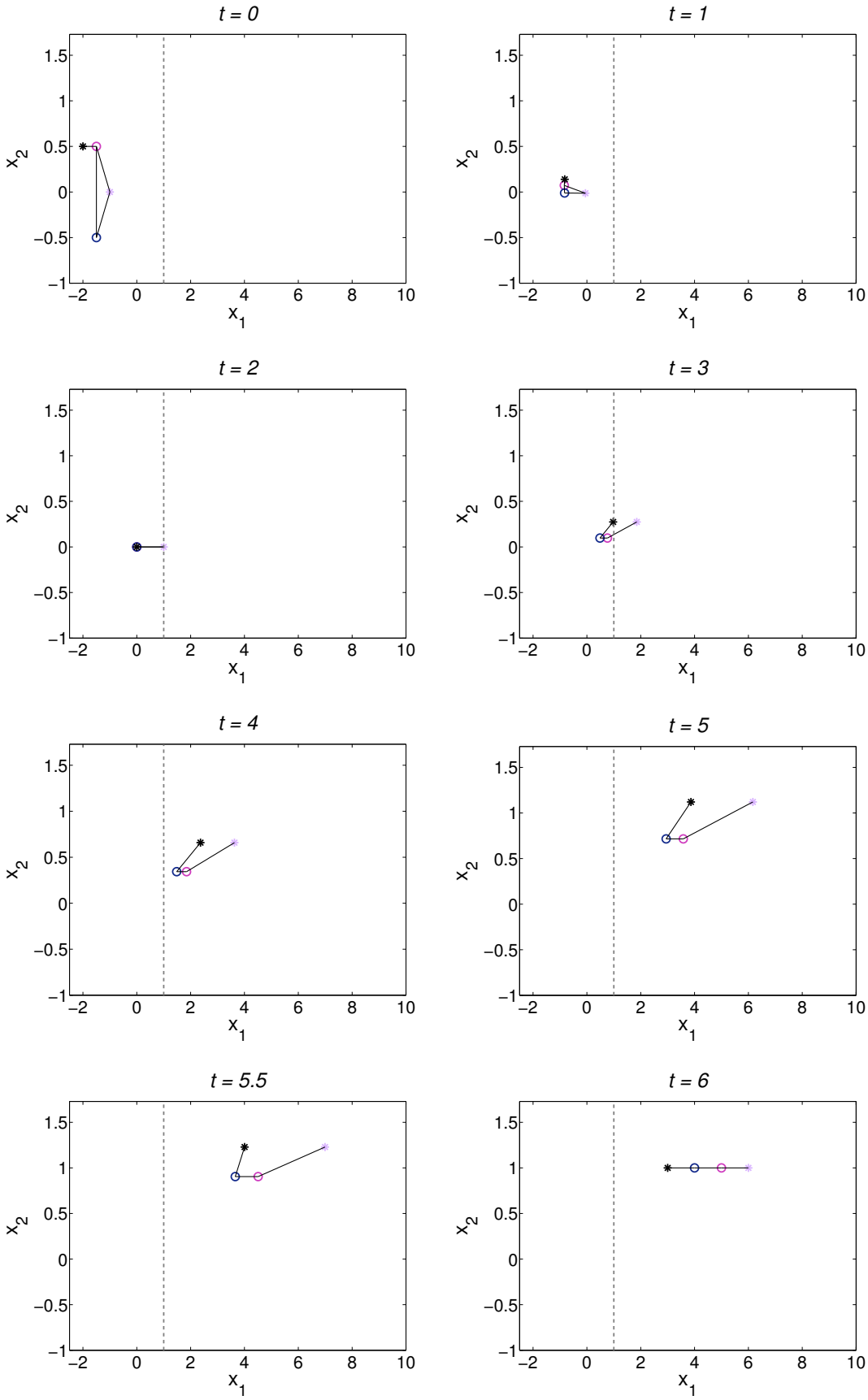
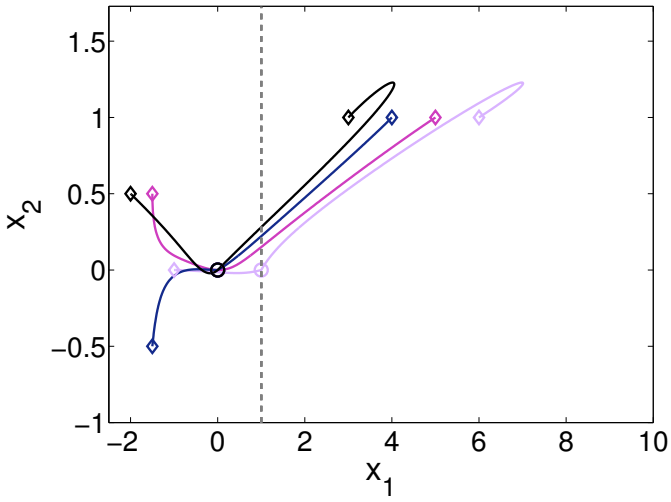
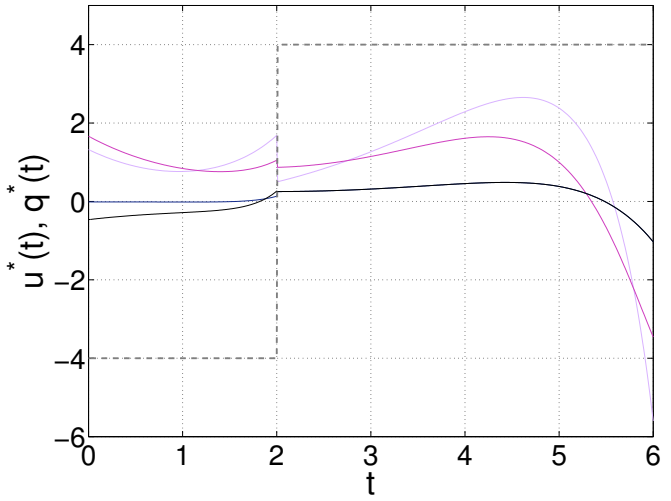


Figure 7.18: Optimal Group Behavior Corresponding To Example 7.5



(a) Optimal States $(x^*_{k,1}(\cdot), x^*_{k,2}(\cdot))$ for each agent k, $k \in \{1, 2, 3, 4\}$



(b) Optimal Input $u^*(\cdot)$

Figure 7.19: Optimal Trajectories Corresponding To Example 7.5

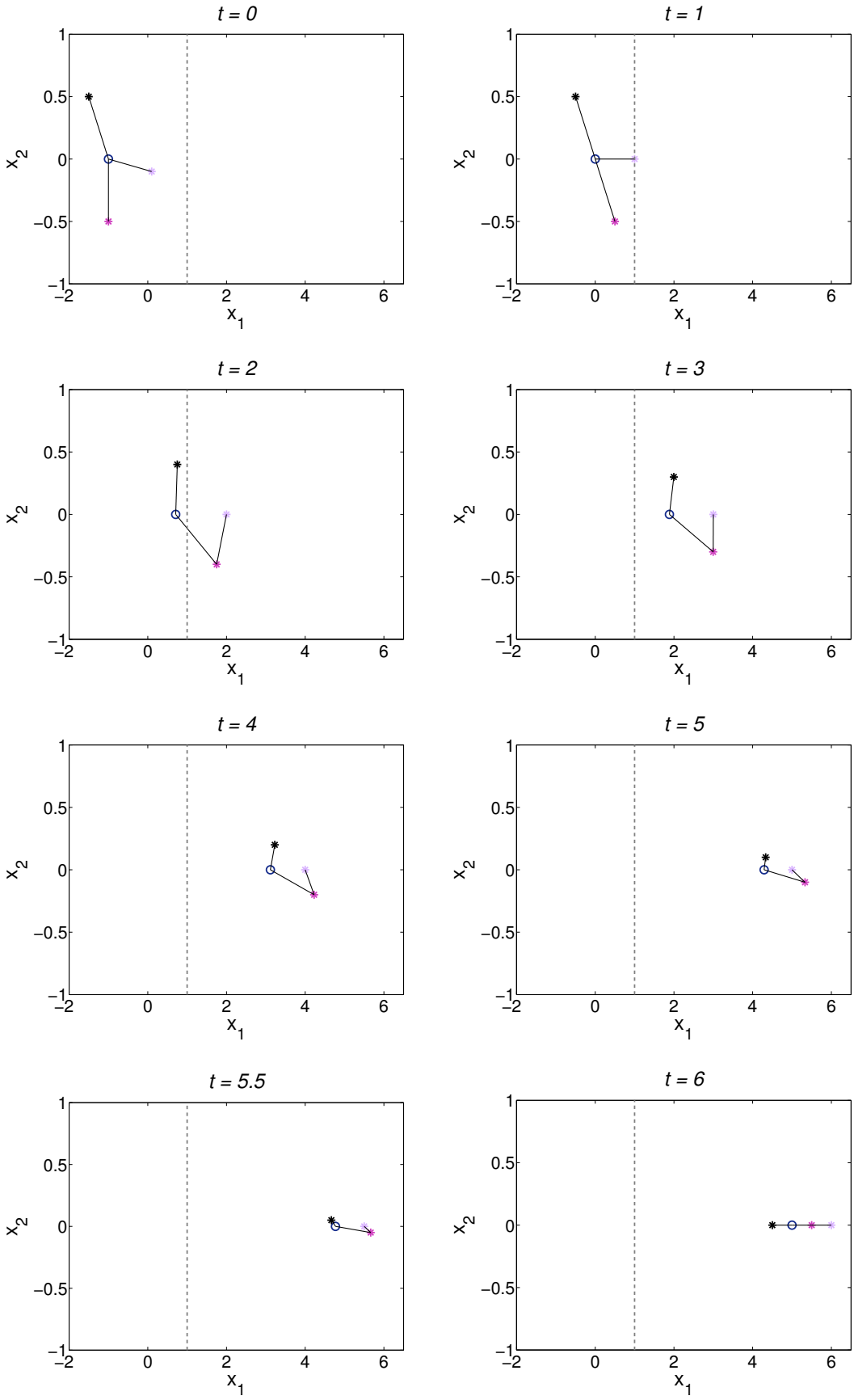
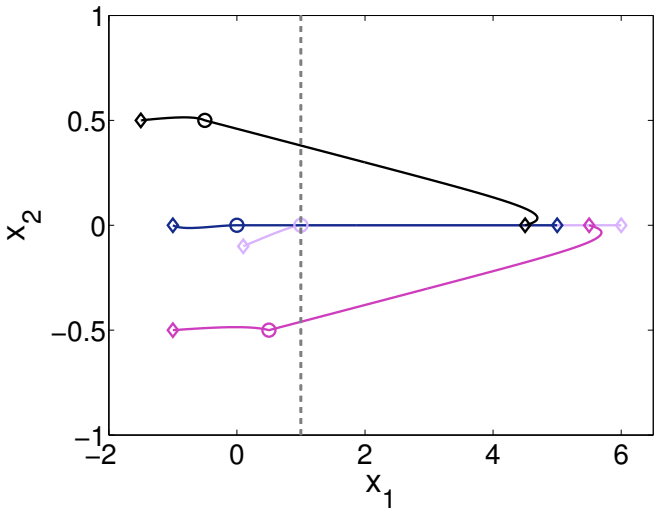
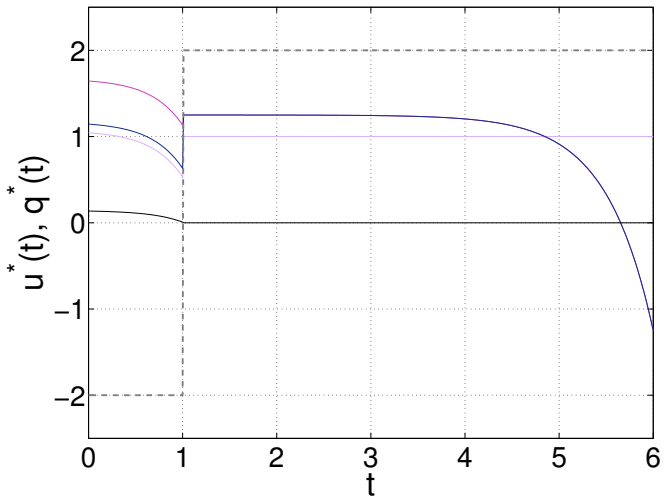


Figure 7.20: Optimal Group Behavior Corresponding To Example 7.6



(a) Optimal States $(x^*_{k,1}(\cdot), x^*_{k,2}(\cdot))$ for each agent k, $k \in \{1, 2, 3, 4\}$



(b) Optimal Input $u^*(\cdot)$

Figure 7.21: Optimal Trajectories Corresponding To Example 7.6

function of the form

$$\ell\left(x(t), u(t)\right) = \sum_{k=1}^{n} \ell_k\left(x_{1,k}; x_{2,k}; \ldots; x_{N_A,k}\right) \tag{7.32}$$

is assumed, it is possible to solve the hybrid optimal control problem separately for each direction k reducing the dimensionality of each subproblem by a factor of n. In the examples of Section 7.4.2, where diagonal matrices Q and R are chosen in (7.15), a division into $n = 2$ subproblems of dimension $k = 4$ is possible.

Nevertheless, besides the restated powerful operation of the proposed hierarchic Dynamic Programming approach, the solution of the hybrid point-to-point problem for this class of systems allows drawing conclusions on the dynamic properties of the assumed leader-follower configuration. For instance, a "tighter" connection between the agents combined with a non-zero Q matrix in (7.15) let the agents move together, as observed in Figure 7.18 in region D_1 (on the left side), where the agents are arranged in a triangle formation. In contrast, in Example 7.6, the agents start in the less connected star formation and, moreover, the agents' states do not appear in the control objective, which is to be minimized. Figure 7.20 and Figure 7.21 show the resulting wide-spread, more "individual" motion of the agents. In fact, the interplay between

(i) the optimal control parameters, that is, the cost function $\ell\left(x(t), u(t)\right)$, the upper bound N on the number of switches, the time horizon T, and the initial and final constraints on the state values $\xi(0) = \xi_0$ and $\xi(T) = \xi_T$,

(ii) the regional-dependent multi-agent dynamics, unambiguously specified by the graph Laplacian $\mathcal{L}$ representing the agents' network topology and the vector κ dividing the group of agents into leaders and followers,

(iii) and the geometric framework, basically given by the switching manifold,

is an interesting field for further studies. Especially noteworthy is that the derived recursive algorithms, cf. Theorem 4.1 and Theorem 5.1, which are used in Section 7.4.2 to analyze the dynamic behavior of multi-agent systems under an optimal control policy, serves also as a design tool answering question like, "Which

agents should be assigned with a leader role in order to achieve the best performance?" or "What is the best way to specify the switching manifolds to obtain the desired group behavior?". In the examples of Section 7.4.2, for instance, the switching behavior depends only on the $x_{1,1}$-state of the first agent. However, a suitable choice may also be the center of a formation.

Finally, it is of particular important to highlight, that the Dynamic Programming approach presented in this work does not require a linear multi-agent dynamic, but is able to deal with more general nonlinear dynamics just as well. This is especially interesting, since the use of such dynamic network models becomes more and more popular not only because they offer a metaphor for representing the reality around us as a world inhabited by autonomous, active, possibly intelligent elements, but also because they provide a methodology enabling the modeling, design, and implementation of large systems in a very modular way. Here, the potential areas of application range from robotics and sensor networks to computer science and artificial intelligence.

CHAPTER VIII

CONCLUSIONS AND OUTLOOK

This chapter concludes the previous investigations on regional dynamics systems by recapitulating the key ideas of the precedent considerations, stating the core results, and highlighting the work's main contributions. This is followed by a brief discussion on possible further extensions and applications of the presented approach.

8.1 Conclusions

This work considers hybrid systems with regional dynamics, i.e. systems, where transitions between different dynamical regimes occur as the system's continuous state reaches given switching surfaces. In other words, the governing dynamics vary depending on the region, the continuous state is evolving in. In particular, the attention is focused on the optimal control problem associated with such systems. More precisely, given a specific cost function, the goal is to determine the optimal path of going from a given starting point to a fixed final state during an *a priori* specified time horizon. A helpful graphical illustration of the problem statement is shown in Figure 3.1.

The key idea and main characteristic of the presented approach is a hierarchical decomposition of the hybrid optimal control problem, which does not only provide a better understanding of the problem's inherent complexity and deliver a deeper insight into the structural composition of the problem, but represents, in fact, the underlying framework of the proposed solution scheme. Moreover, the hierarchical structure, composed of different levels of abstraction, makes it possible to apply an adequate Dynamic Programming algorithm and obtain, finally, a characterization of global optimality, given an upper bound on the number of discrete transition

along a hybrid trajectory. The way this is actually done is by approaching and solving the introduced multimodal point-to-point problem on three different levels of control: On the highest level, the geometric framework is taken into account. A discrete representation, the so-called transition automaton, is introduced, which specifies the connections between the different regions and the associated language provides all sequences of transitions, which are possible in the given partitioned state space. At a level below, the main theorem of this work, the Hybrid Bellman Equation for systems with regional dynamics, is stated. This recursive algorithm is based on both, the global accessibility conditions provided by the automaton on the highest level of control and the costs computed on the lowest level of control, which, in fact, represent the solution to standard (non-hybrid) state-constrained optimal control problems calculated, in each case, between two points on the switching manifolds. Not surprisingly, the optimal solution is hybrid in nature in that it depends on not only the continuous control signals, but also on discrete decisions as to what domains the system should go through in the first place.

The main benefit with the proposed approach lies in the fact that a hierarchical Dynamic Programming algorithm can be used to representing both a mathematical formulation and theoretical characterization of the hybrid solution's structural composition and, from a more application-driven point of view, an implementable, numerically computable calculation rule. On the one hand, based on the hierarchical structuring, the derived Hybrid Bellman Equation captures the problem's complexity and provides doubtlessly a beneficial tool in understanding, analyzing, and designing regional dynamics systems. For example, the reachability issue in the multimodal point-to-point problem can be investigated on the high-level languages generated by the transition automaton. Furthermore, from a design perspective, on this level of abstraction, a desired switching behavior can be enforced by incorporating the appropriate switching rules in the transition automaton. On the other hand, in a computational context, the hierarchic organization of the

problem turns out to be again a helpful structure indicating, this time, the numerical way of proceeding. The computability and practicability of the recursive algorithm is proven by a number of examples. Especially, the numerical solution of a multi-agent problem highlights the powerful operation of the proposed approach and results in some insightful simulations of the agents' interaction behavior.

Finally, special emphasis should be placed on the universal validity of the hierarchic Dynamic Programming approach, which can be applied to a significantly larger class of hybrid optimal control problems, cf. Chapter 6.

In brief, the novelty of the solution herein lies in the treatment of global optimality conditions for the general class of regional dynamics systems through a Dynamic Programming approach, where the control variable consists not only of the continuous control signal, but also of a discrete decision variable dictating what regions the system should switch between. A number of examples illustrate the use of the proposed method.

8.2 Outlook

Recommendations for future research include both,

(i) theoretical extensions and mathematical generalizations of the hierarchic Dynamic Programming approach

(ii) and more application-oriented improvements of the algorithm's computational implementation.

In the latter category fall further efforts on producing an usable, helpful, and more general software tool allowing an easy input of the model specifications and a sufficiently fast solution of the hybrid point-to-point problem. This means, in particular, that, in the case of application-driven examples, Assumption 5.1 is not necessarily satisfied and the simplified calculation of the cost functions $c(\cdot, \cdot, \cdot, \cdot, \cdot)$ may not be possible anymore. Thus, an important step to a user-friendly application tool is the implementation of a (non-hybrid) state-constrained optimization

algorithm for the calculation of $c(\cdot, \cdot, \cdot, \cdot, \cdot)$, cf. [7,17,21,22,26,30,53]. The computational complexity of this optimization algorithm has, of course, a direct impact on the effort associated with the overall hybrid optimal control solver. Thus, the decision on which calculation rule shows the best performance in the considered multiregional context is to be made very carefully. Furthermore, not implemented so far, is an algorithm which allows to enter the transition automaton or even the geometric framework itself and generates the corresponding languages. Furthermore, as indicated in Section 5.7, the precise dependence of the numerical effort on the regional geometric structure is fairly complex and represents a fascinating topic for further studies.

As regards the theoretical extensions of the HOCP formulation, a huge number of ideas and the adequate changes to be made in the previous hierarchic Dynamic Programming approach are listed in Chapter 6. However, concerning the presented generalizations, the preliminary investigations in Chapter 6 should be extended by more detailed definitions, an appropriate literature overview, examples for applications, and numerical simulations.

In the end, the discussions in this section reflect the remarkable characteristic of the derived hierarchic Dynamic Programming approach in that it represents both a result of theoretical and computational relevance.

REFERENCES

[1] Anderson, B. D. O. and Moore, J. B., *Linear Optimal Control*. Prentice Hall, 1971.

[2] Ando, H., Oasa, Y., Suzuki, I., and Yamashita, M., "Distributed memoryless point convergence algorithm for mobile robots with limited visibility," *IEEE Transactions on Robotics and Automation*, vol. 15, pp. 818–828, 1999.

[3] Athans, M. and Falb, P., *Optimal Controls (Electrical & Electronic Engineering)*. Mcgraw-Hill College, 1966.

[4] Bellman, R., *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957.

[5] Bellman, R. E. and Dreyfus, S. E., *Applied Dynamic Programming*. Princeton University Press, 1962.

[6] Bemporad, A., Borrelli, F., and Morari, M., "On the optimal control law for linear discrete time hybrid systems," in *Hybrid Systems: Computation and Control*, vol. 2289 of *Lecture Notes in Computer Science*, pp. 105–119, Springer-Verlag, 2002.

[7] Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E. N., "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[8] Bjorkenstam, S., Ji, M., Egerstedt, M., and Martin, C., "Leader-based multi-agent coordination through hybrid optimal control," in *Allerton Conference on Communication, Control, and Computing*, 2006.

[9] Boltyanskii, V. G., *Optimal Control of Discrete Systems*. John Wiley, 1978. Translated from Russian.

[10] Borrelli, F., *Constrained Optimal Control of Linear and Hybrid Systems (Lecture Notes in Control and Information Sciences).* No. 290, Springer-Verlag, 2003.

[11] Branicky, M., Borkar, V., and Mitter, S., "A unified framework for hybrid control: model and optimal control theory," *IEEE Transactions on Automatic Control*, vol. 43, no. 1, pp. 31–45, 1998.

[12] Branicky, M. and Mitter, S., "Algorithms for optimal hybrid control," in *Proceedings of the 34th IEEE Conference on Decision and Control*, vol. 3, pp. 2661–2666, 1995.

[13] Bryson, A. E. and Ho, Y.-C., *Applied Optimal Control: Optimization, Estimation, & Control.* Taylor & Francis, 1975.

[14] Caines, P., Egerstedt, M., Malhamé, R., and Schöllig, A., "A hybrid Bellman equation for bimodal system," in *Hybrid Systems: Computation and Control*, vol. 4416 of *Lecture Notes in Computer Science*, pp. 656–659, Springer-Verlag, 2007.

[15] Caines, P. E. and Shaikh, M. S., "Optimality zone algorithms for hybrid systems: Efficient algorithms for optimal location and control computation," in *Hybrid Systems: Computation and Control*, vol. 3927 of *Lecture Notes in Computer Science*, pp. 123–137, Springer-Verlag, 2006.

[16] Cassandras, C. G. and Lafortune, S., *Introduction to Discrete Event Systems (The International Series on Discrete Event Dynamic Systems).* Springer-Verlag, 1999.

[17] Chryssoverghi, I., Coletsos, I., and Kokkinis, B., "Discretization methods for optimal control problems with state constraints," *Journal of Computational and Applied Mathematics*, vol. 191, pp. 1–31, 2006.

[18] Chui, C. K. and Chen, G., *Linear Systems and Optimal Control*. Springer-Verlag, 1989.

[19] Desai, J., Ostrowski, J. P., and Kumar, V., "Controlling formations of multiple mobile robots," in *IEEE International Conference on Robotics and Automation*, pp. 2864–2869, 1998.

[20] Desai, J., Ostrowski, J., and Kumar, V., "Modeling and control of formations of nonholonomic mobile robots," *IEEE transactions on Robotics and Automation*, vol. 17, no. 6, pp. 905–908, 2001.

[21] Dontchev, A. L. and Hager, W. W., "The euler approximation in state constrained optimal control," *Mathematics of Computation*, vol. 70, pp. 173–203, 2001.

[22] Egerstedt, M. and Martin, C. F., "Optimal control and monotone smoothing splines," in *New Trends in Nonlinear Dynamics and Control and their Applications*, vol. 295 of *Lecture Notes in Control and Information Sciences*, pp. 279–294, Springer-Verlag, 2004.

[23] Egerstedt, M., Wardi, Y., and Axelsson, H., "Transition-time optimization for switched-mode dynamical systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 110–115, 2006.

[24] Fax, J. A. and Murray, R. M., "Graph Laplacian and stabilization of vehicle formations," in *Proceedings of the 15th IFAC Conference*, pp. 283–288, 2002.

[25] Fax, J. and Murray, R., "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1465–1476, Sept 2004.

[26] Gerdts, M., "Global convergence of a nonsmooth newton's method for control-state constrained optimal control problems," *Hamburger Beitrge zur Angewandten Mathematik*, no. 2007-04, 2007.

[27] Ghosh, M. K., Arapostathis, A., and Marcus, S. I., "Optimal control of switching diffusions with application to flexible manufacturing systems," *SIAM Journal on Control and Optimization*, vol. 31, no. 5, pp. 1183–1204, 1993.

[28] Giua, A., Seatzu, C., and Van der Mee, C., "Optimal control of switched autonomous linear systems," in *Proceedings of the 40th IEEE Conference on Decision and Control*, vol. 3, pp. 2472–2477, 2001.

[29] Godsil, C. and Royle, G., *Algebraic graph theory*. Springer, 2001.

[30] Goodwin, G. C., Seron, M. M., and De Doná, J. A., *Constrained Control and Estimation: An Optimisation Approach (Communications and Control Engineering)*. Springer-Verlag, 2004.

[31] Heagy, J. F., Carroll, T. L., and Pecora, L. M., "Synchronous chaos in coupled oscillator systems," *Physical Review E*, vol. 50, no. 3, pp. 1874–1885, 1994.

[32] Hedlund, S. and Rantzer, A., "Optimal control of hybrid systems," in *Proceedings of the 38th IEEE Conference on Decision and Control*, vol. 4, pp. 3972–3977, 1999.

[33] Jadbabaie, A., Lin, J., and Morse, A. S., "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, pp. 988–1001, June 2003.

[34] Ji, M. and Egerstedt, M., "Distributed coordination control of multi-agent systems while preserving connectedness," *IEEE Transactions on Robotics*, 2007. To appear.

[35] Ji, M., Muhammad, A., and Egerstedt, M., "Leader-based multi-agent coordination: Controllability and optimal control," in *American Control Conference*, pp. 1358–1363, 2006.

[36] Ji, M., *Graph-Based Control of Networked Systems*. PhD thesis, School of Electrical andComputer Engineering, Georgia Institute of Technology, Atlanta, 2007.

[37] Kirk, D. E., *Optimal Control Theory: An Introduction*. Dover Publications, 2004.

[38] Knowles, G., *An Introduction to Applied Optimal Control (Mathematics in Science and Engineering)*, vol. 159. Academic Press, 1982.

[39] Lee, E. B. and Markus, L., *Foundations of Optimal Control Theory*. Krieger Pub Co, 1986.

[40] Lin, J., Morse, A., and Anderson, B., "The multi-agent rendezvous problem," in *Proceedings of the* 42^{nd} *IEEE Conference on Decision and Control*, pp. 1508–1513, 2003.

[41] Lin, Z., Broucke, M., and Francis, B., "Local control strategies for groups of mobile autonomous agents," *IEEE Transactions on Automatic Control*, vol. 49, no. 4, pp. 622–629, 2004.

[42] Locatelli, A., *Optimal Control: An Introduction*. Birkhäuser Basel, 2001.

[43] Luenberger, D. G., *Optimization By Vector Space Methods*. Wiley, 1969.

[44] Malhamé, R. and Chong, C.-Y., "Electric load model synthesis by diffusion approximation of a high-order hybrid-state stochastic system," *IEEE Transactions on Automatic Control*, vol. 30, no. 9, pp. 854–860, 1985.

[45] Mesbahi, M., "State-dependent graphs," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, pp. 3058–3063, 2003.

[46] Olfati-Saber, R., "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[47] Olfati-Saber, R. and Murray, R. M., "Distributed structural stabilization and tracking for formations of dynamic multi-agents," in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 1, pp. 209–215, 2002.

[48] Olfati-Saber, R. and Murray, R. M., "Agreement problems in networks with directed graphs and switching toplogy," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 4, pp. 4126–4132, 2003.

[49] Olfati-Saber, R. and Murray, R. M., "Flocking with obstacle avoidance: cooperation with limited communication in mobile networks," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 2, pp. 2022–2028, 2003.

[50] Pecora, L. M. and Carroll, T. L., "Master stability functions for synchronized coupled systems," *Physical Review Letters*, vol. 80, no. 10, pp. 2109–2112, 1998.

[51] Pinch, E. R., *Optimal Control and the Calculus of Variations*. Oxford University Press, USA, 2002.

[52] Pontryagin, L. S., Boltyanski, V. G., Gamkrilidze, R. V., and Mischenko, E. F., *The Mathematical Theory of Optimal Processes*. Interscience Publishers, New York, 1962.

[53] Pytlak, R., *Numerical Methods for Optimal Control Problems with State Constraints (Lecture Notes in Mathematics)*. Springer-Verlag, 2006.

[54] Rantzer, A. and Johansson, M., "Piecewise linear quadratic optimal control," *IEEE Transactions on Automatic Control*, vol. 45, no. 4, pp. 629–637, 2000.

[55] Rao, S. S., *Engineering Optimization: Theory and Practice, 3rd Edition*. Wiley-Interscience, 1996.

[56] Ren, W. and Beard, R., "Consensus of information under dynamically changing interaction topologies," in *Proceedings of the American Control Conference 2004*, vol. 6, pp. 4939–4944, 2004.

[57] Reynolds, C., "Flocks, herds, and schools: A distributed behavioral model," *Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987.

[58] Schöllig, A., Caines, P. E., Egerstedt, M., and Malhamé, R. P., "A hybrid Bellman equation for systems with regional dynamics," in *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007. To appear.

[59] Shaikh, M. S., *Optimal control of hybrid systems: Theory and algorithms*. PhD thesis, Department of Electrical and Computer Engineering, McGill University, Montréal, Canada, 2004.

[60] Shaikh, M. S. and Caines, P. E., "On trajectory optimization for hybrid systems: theory and algorithms for fixed schedules," in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 2, pp. 1997–1998, 2002.

[61] Shaikh, M. S. and Caines, P. E., "On the optimal control of hybrid systems: Optimization of trajectories, switching times and location schedules," in *Hybrid Systems: Computation and Control, LNCS 2623*, pp. 466–481, 2003.

[62] Shaikh, M. S. and Caines, P. E., "On the hybrid optimal problem: Theory and algorthims," *IEEE Transactions on Automatic Control*, 2007. To appear.

[63] Shaikh, M. and Caines, P., "Optimality zone algorithms for hybrid systems computation and control: From exponential to linear complexity," in *Proceedings of the 44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference*, pp. 1403–1408, 2005.

[64] Sussmann, H., "Set-valued differentials and the hybrid maximum principle," in *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 1, pp. 558–563, 2000.

[65] Tanner, H., Jadbabaie, A., and Pappas, G., "Stable flocking of mobile agents, part II : Dynamic topology," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, pp. 2016–2021, 2003.

[66] Tanner, H., Pappas, G., and Kumar, V., "Leader-to-formation stability," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 443–455, 2004.

[67] Vicsek, T., Czirok, A., Jacob, E. B., Cohen, I., and Schochet, O., "Novel type of phase transitions in a system of self-driven particles," *Physical Review Letters*, vol. 75, pp. 1226–1229, 1995.

[68] Witsenhausen, H., "A class of hybrid-state continuous-time dynamic systems," *IEEE Transactions on Automatic Control*, vol. 11, no. 2, pp. 161–167, 1966.

[69] Xu, X. and Antsaklis, P., "Optimal control of switched autonomous systems," in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 4, pp. 4401–4406, 2002.